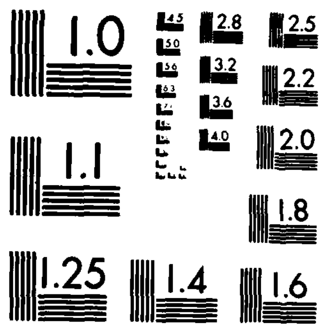
AD-A153 612 A COMPARISON OF METHODS OF LEAST SQUARES ADJUSTMENT OF TRAVERSES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA S AUMCHANTR DEC 84 1/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

AD-A153 612

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Comparison of Methods
of
Least Squares Adjustment of Traverses

by

Saman Aumchantr

December 1984

Thesis Advisor: Rolland L. Hardy

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Comparison of Methods of Least Squares Adjustment of Traverses		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Saman Aumchantr		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1984
		13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if d' .erent from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adjustment, Approximate Method, Comparison, Condition Equation Method, Least Squares Method, Traverse, UTM grid coordinates.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Traverse is a method of surveying in which a sequence of lengths and directions of lines between points on the Earth are measured and used in determining positions of the points. This method is one of several used to find the accurate geodetic positions which various agencies use. Traversing is a convenient, rapid method for establishing horizontal control.

The theoretical background is provided here to explain the method of traverse station position computations and adjustments in the Universal

Block 20 (continued)

Transverse Mercator grid coordinates. Closed traverse station positions were computed and adjusted using the Approximate Method and by the Least Squares Method. The adjusted coordinates of both methods were transformed from the Universal Transverse Mercator grid coordinates to geographic coordinates and compared with the coordinates which were adjusted by the U.S. National Ocean Service.

Approved for public release; distribution unlimited.

A Comparison of Methods of Least Squares Adjustment of Traverses

by

Saman Aumchantr
Lieutenant, Royal Thai Navy
B.S., Royal Thai Naval Academy, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN HYDROGRAPHIC SCIENCES

from the

NAVAL POSTGRADUATE SCHOOL
December 1984

Author: Saman Aumchantr
Saman Aumchantr

Approved by: Rolland R. Hardy
Rolland L. Hardy, Thesis Advisor

Glen R. Schaefer
Glen R. Schaefer, Second Reader

Christopher N.K. Mooers
Christopher N.K. Mooers, Chairman, Department of Oceanography

J N Dyer
John N. Dyer, Dean of Science and Engineering

ABSTRACT

Traverse is a method of surveying in which a sequence of lengths and directions of lines between points on the Earth are measured and used in determining positions of the points. This method is one of several used to find the accurate geodetic positions which various agencies use. Traversing is a convenient, rapid method for establishing horizontal control.

The theoretical background is provided here to explain the method of traverse station position computations and adjustments in the Universal Transverse Mercator grid coordinates. Closed traverse station positions were computed and adjusted using the Approximate Method and by the Least Squares Method. The adjusted coordinates of both methods were transformed from the Universal Transverse Mercator grid coordinates to geographic coordinates and compared with the coordinates which were adjusted by the U.S. National Ocean Service.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I express sincere gratitude to my Thesis Advisor, Dr. Rolland L. Hardy, and Second Reader, Cdr. Glen R. Schaefer, for their suggestions and assistance. Finally, I thank Ms. Tamara M. Hayling, Lt. Nicholas E. Perugini, and Messrs. Mark L. Faye, Peter J. Rakowsky, and James R. Cherry who made this thesis possible.

I. INTRODUCTION

Hydrographic surveying includes many branches of science for the purpose of the production of nautical charts specially designed for use by the mariner. The determination of position in hydrographic surveying is as important as the measurement of depth. Before determining an accurate hydrographic position, accurate geodetic positions[1] for shore control must be established. There are many methods available to establish geodetic control in the survey area. These methods are:

1. Triangulation
2. Trilateration
3. Traverse
4. Intersection
5. Resection

The process of making a proper nautical chart consists, first of all, in setting up a framework of marks on the ground. Before 1950 the main framework of a geodetic survey almost always consisted of triangulation, which was replaced by traverse if the topography made triangulation impracticable. During the last decade, the introduction of electronic distance measuring (EDM) equipment has made both trilateration and traverse economical, and an acceptable substitute for triangulation. In fact, it appears probable that these new methods will replace triangulation as the main framework for new geodetic surveys [Ref. 1]. It is evident that triangulation and traverse are the main methods used for establishing control. There seems to be no

---

[1]A position of a point on the surface of the Earth expressed in terms of geodetic latitude, geodetic longitude, and geodetic height. A geodetic position implies an adopted geodetic datum.

agreement among the various agencies as to which of these two methods is mostly used. The U.S. National Ocean Service (NOS) does the majority of its horizontal control surveys for hydrography with traverse (about 90%) [Ref. 2]. The main factor for the selection of one or the other method depends on the geographical configuration of the survey project area and the availability of good EDM equipment. Traversing is a convenient, rapid method for establishing horizontal control. It is particularly useful in densely built areas, when the coastline tends to be even, along a railroad track, and in heavily forested areas where lengths of sight are short so that neither triangulation nor trilateration is suitable.

The objectives of this thesis are to show methods of traverse station position computation and a comparison of methods of least squares adjustment. Closed traverse station positions were computed and adjusted using the Approximate Method and by the Least Squares Method in the Universal Transverse Mercator (UTM) grid coordinates. The computer programs were written to calculate the adjusted traverse station positions. Test data included those data obtained during the Geodetic Survey Field Experience course at the Naval Postgraduate School (NPS) in October 1983.

The results of comparative computations are shown to more significant figures, in this thesis, than are normally considered desirable in production work. The same observed data are used with each of three computational methods. It is important to recognize that what is being compared is not observational precision but computational precision. Hence, it is considered necessary for a rigorous comparison of computational precision, including round off error, to show results to several more decimal places than is justifiable based on observational precision alone.

II. TRAVERSE

A. GENERAL

Traverse is a method of surveying in which a series of straight lines connect successive established points along the route of a survey. An angular measurement is taken using a theodolite at each point where the traverse changes direction. Distances along the line between successive traverse points are determined by EDM equipment. The points defining the ends of the traverse lines are called turning points, traverse points, or traverse stations. Each straight section of a traverse is called a leg or a traverse line.

A closed traverse originates at a point of known position and is closed on another point of known horizontal position. Traverse I-1-2-3-F originates at point I with a backsight along line IA of known azimuth and closes on point F, with a foresight along line FB also of known azimuth (Figure 2.1). This type of traverse is preferable to all

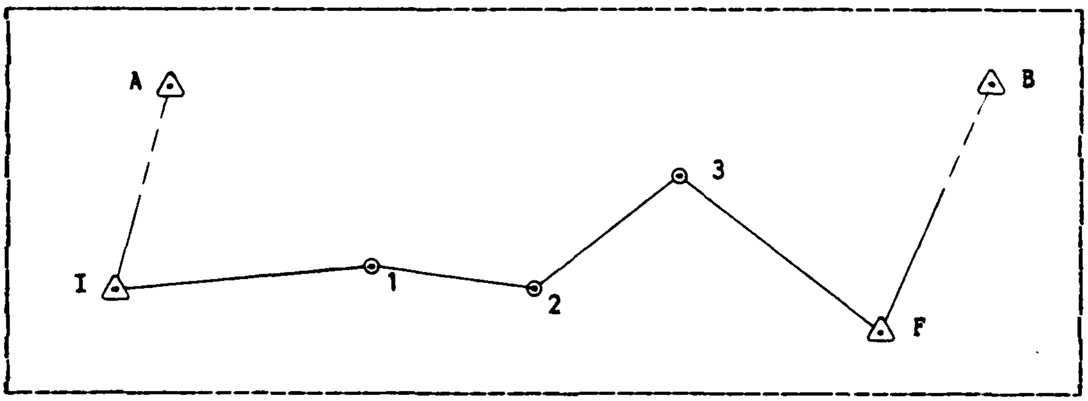


Figure 2.1 Closed Traverse.

others since computational checks are possible which allow detection of systematic errors[2] in both distance and direction.

A closed-loop (closed polygon) traverse is a special case of a closed traverse in which the originating and terminating points are the same point with a known horizontal position. Traverse I-1-2-3-4-I, originates and terminates on point I (Figure 2.2). This type of traverse permits an internal check on the angles, but there is no check on the linear measurements. Therefore, there is a possibility that an error proportional to distance may occur and not be detected.

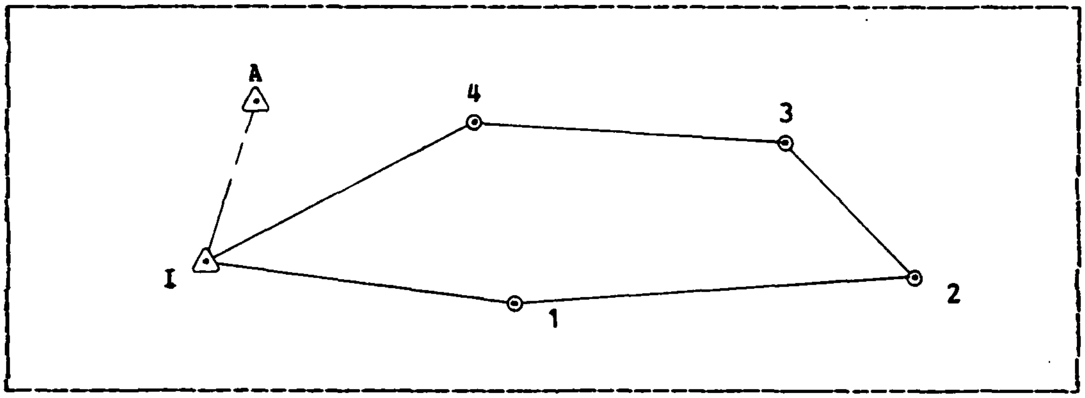


Figure 2.2 Closed-loop Traverse.

B. ANGLE AND DIRECTION MEASUREMENT

Angles and directions may be defined by means of bearings, azimuths, deflection angles, angles to the right, or interior angles. These quantities are said to be observed

---

[2]Systematic errors may be caused by faulty instruments or factors such as temperature or humidity changes which affect the performance of measuring instruments.

when obtained directly in the field and calculated when obtained indirectly by computation.

A theodolite is an instrument designed to observe horizontal directions and measure vertical angles. It consists of a telescope mounted to rotate vertically on a horizontal axis supported by a pair of vertical standards attached to a revolvable circular plate containing a graduated circle for observing horizontal directions. Another graduated arc is attached to one standard so that vertical angles can be measured.

Repeating and direction theodolites have features that are common to both types of instruments. Repeating theodolites are read directly to 20" or 01' and by estimation to one-tenth the corresponding direct reading. Direction theodolites are usually read directly to 01" and can be estimated to tenths of seconds [Ref. 3, p. 215]. In general, direction theodolites are more precise than are repeating theodolites.

The direction theodolite observes directions only and angles are computed by subtracting one direction from another. Assume that the horizontal angle AIB (Figure 2.3) is to be measured with a direction theodolite. The theodolite is set over point I, leveled and centered, and a sight is taken on point A. The horizontal circle is then viewed through the optical-viewing system and the circle reading is observed and recorded. Assume the reading is 45° 02' 40". The telescope is then sighted on point B. The horizontal circle is then viewed through the optical-viewing system and the circle reading is observed and recorded. Assume the reading is 124° 11' 59". These two observations constitute directions which have a common reference direction that is completely arbitrary. The clockwise horizontal angle is

$$i = (124° \; 11' \; 59") - (45° \; 02' \; 40") = 79° \; 09' \; 19"$$

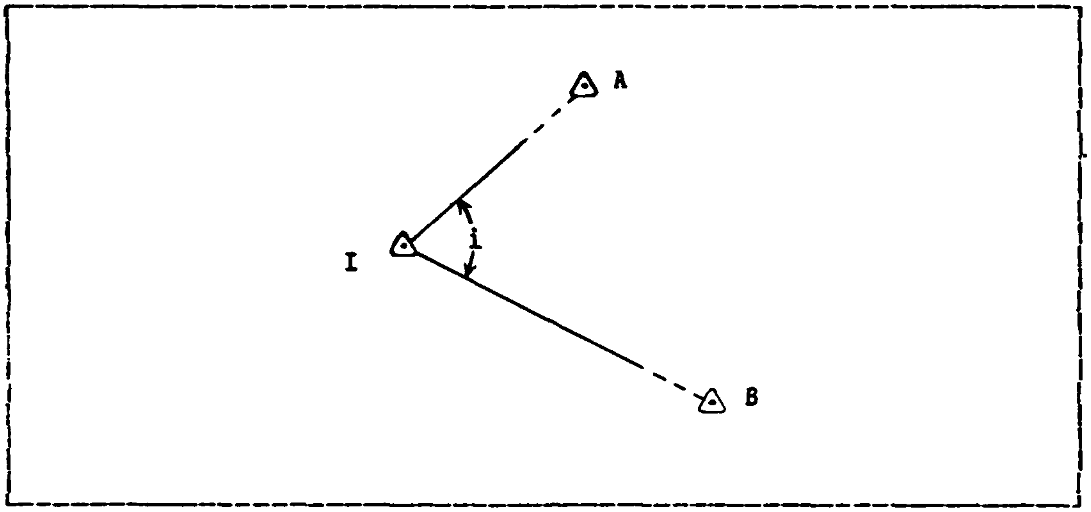


Figure 2.3 Horizontal Angle.

C. DISTANCE MEASUREMENTS

There are several methods of determining distance, the choice of which depends on the accuracy required and the cost. For example, tacheometry, taping, and EDM equipment can be used. The general availability of EDM equipment has practically eliminated the use of taping for the measurement of traverse lengths. Accuracies are comparable, or superior, to those obtained with invar tapes.

Distances measured using EDM equipment are subject to errors arising from the instrumental components, calibration of the equipment, inaccuracies in the meteorological data, elevation discrepancies, and centering of the instruments or reflectors. The reduction of measured distances involves converting the slope distance to a horizontal distance, converting the chord distance to an equivalent arc distance, and reducing the arc distance to the ellipsoid. The reduction of slope distance to horizontal distance is necessary to compensate for the difference in elevation of the end points of the measured line. The horizontal distance is

reduced to the geodetic distance by applying a sea level corrector and a chord-arc corrector to the horizontal distance [Ref. 4, pp. 124-125].

D. ACCURACY

In general, the accuracy of a traverse is judged on the basis of the resultant error of closure of the traverse. This resultant closing error is a function of the accuracies in the measurement of directions and distances. The classification, standards of accuracy, and general specifications for horizontal control have been prepared by the Federal Geodetic Control Committee (FGCC) [Ref. 5] and have been reviewed by the American Society of Civil Engineers, The American Congress on Surveying and Mapping, and the American Geophysical Union (Table I). The third-order traverse class I and class II are of particular interest because these are the orders of accuracy the hydrographer is usually required to accomplish.

E. DATA ACQUISITION

The data for this thesis were acquired at Moss Landing, California, by NPS Hydrographic students during the Geodetic Survey Field Experience course in October 1983. All of the known station positions and azimuths (Table II) were adjusted by the Coast and Geodetic Survey by third-order methods [Ref. 6].

The distances were observed by using a Kern DM102 and a Tellurometer MRA5. The Kern DM102 is an electro-optical distance meter. The measuring accuracy was ±(5 mm + 5 ppm). The Tellurometer MRA5 is a microwave instrument. Precision in terms of probable error, in the temperature range of -32°C to +44°C, is ±(5 cm + 100 ppm) for a single

TABLE I

Classification, Standards of Accuracy, and General Specifications for Horizontal Control

Classification	TRAVERSE First-Order	Second-Order Class I	Second-Order Class II	Third-Order Class I	Third-Order Class II
Recommended spacing of principal stations	Network stations 10-15 km Other surveys seldom less than 3 km.	Principal stations seldom less than 4 km except in metropolitan area surveys where the limitation is 0.3 km.	Principal stations seldom less than 2 km except in metropolitan area surveys where the limitation is 0.2 km.	Seldom less than 0.1 km in tertiary surveys in metropolitan area surveys. As required for other surveys.	
Horizontal directions or angles *					
Instrument	0".2	0".2 or 1".0	0".2 or 1".0	1".0	1".0
Number of observations	16	8 or 12*	6 or 8*	4	2
Rejection limit from mean	4"	4" or 5"	4" or 5"	5"	5"
Length measurements					
Standard error [1]	1 part in 600,000	1 part in 300,000	1 part in 120,000	1 part in 60,000	1 part in 30,000
Reciprocal vertical angle observations *					
Number of and spread between observations	3 D/R—10"	3 D/R—10"	2 D/R—10"	2 D/R—10"	2 D/R—20"
Number of stations between known elevations	4-6	6-8	8-10	10-15	15-20
Astro azimuths					
Number of courses between azimuth checks [7]	5-6	10-12	15-20	20-25	30-40
No. of obs./night	16	16	12	8	4
No. of nights	2	2	1	1	1
Standard error	0".45	0".45	1".5	3".0	8".0
Azimuth closure at azimuth check point not to exceed *	1".0 per station or 2" $\sqrt{N}$	1".5 per station or 3" $\sqrt{N}$. Metropolitan area surveys seldom to exceed 2".0 per station or 3" $\sqrt{N}$	2".0 per station or 6" $\sqrt{N}$. Metropolitan area surveys seldom to exceed 4".0 per station or 8" $\sqrt{N}$	3".0 per station or 10" $\sqrt{N}$. Metropolitan area surveys seldom to exceed 6".0 per station or 15" $\sqrt{N}$	8" per station or 30" $\sqrt{N}$
Position closure ** after azimuth adjustment	0.04m $\sqrt{K}$ or 1:100,000	0.08 $\sqrt{K}$ or 1:50,000	0.2m $\sqrt{K}$ or 1:20,000	0.4m $\sqrt{K}$ or 1:10,000	0.8m $\sqrt{K}$ or 1:5,000

* May be reduced to 8 and 4, respectively, in metropolitan areas.

TABLE I
(Continued)

NOTE (1)

The standard error is to be estimated by

$$\sigma_m = \sqrt{\frac{\Sigma v^2}{n(n-1)}}$$

where σ_m is the standard error of the mean, v is a residual (that is, the difference between a measured length and the mean of all measured lengths of a line), and n is the number of measurements.

The term "standard error" used here is computed under the assumption that all errors are strictly random in nature. The true or actual error is a quantity that cannot be obtained exactly. It is the difference between the true value and the measured value. By correcting each measurement for every known source of systematic error, however, one may approach the true error. It is mandatory for any practitioner using these tables to reduce to a minimum the effect of all systematic and constant errors so that real accuracy may be obtained. (See page 267 of Coast and Geodetic Survey Special Publication No. 247, "Manual of Geodetic Triangulation," Revised edition, 1959, for definition of "actual error.")

NOTE (2)

The figure for "Instrument" describes the theodolite recommended in terms of the smallest reading of the horizontal circle. A position is one measure, with the telescope both direct and reversed, of the horizontal direction from the initial station to each of the other stations. See FGCC "Detailed Specifications" for number of observations and rejection limits when using transits.

NOTE (3)

The standard error for astronomic azimuths is computed with all observations considered equal in weight (with 75 percent of the total number of observations required on a single night) after application of a 5 second rejection limit from the mean for First- and Second-Order observations.

NOTE (4)

See FGCC "Detailed Specifications" on "Elevation of Horizontal Control Points" for further details. These elevations are intended to suffice for computations, adjustments, and broad mapping and control projects, not necessarily for vertical network elevations.

NOTE (5)

Unless the survey is in the form of a loop closing on itself, the position closures would depend largely on the constraints or established control in the adjustment. The extent of constraints and the actual relationship of the surveys can be obtained through either a review of the computations, or a minimally constrained adjustment of all work involved. The proportional accuracy or closure (i.e. 1/100,000) can be obtained by computing the difference between the computed value and the fixed value, and dividing this quantity by the length of the loop connecting the two points.

NOTE (6)

See FGCC "Detailed Specifications" on "Trilateration" for further details.

NOTE (7)

The number of azimuth courses for First-Order traverses are between Laplace azimuths. For other survey accuracies, the number of courses may be between Laplace azimuths and/or adjusted azimuths.

NOTE (8)

The expressions for closing errors in traverses are given in two forms. The expression containing the square root is designed for longer lines where higher proportional accuracy is required.

The formula that gives the smallest permissible closure should be used.

N is the number of stations for carrying azimuth.
K is the distance in kilometers.

TABLE II

Coordinates of Known Stations and Azimuths

Station	UTM grid coordinates Grid Northing (m.)	Grid Easting (m.)
Moss 2	4,072,555.85206	608,279.04404
Holm	4,079,258.31754	612,238.85256

Grid azimuth clockwise from North				Azimuth °	'	"
From Moss 2	to	Pipher	stations	100	16	23.778
From Holm	to	Moran	stations	136	33	26.334

determination. The distances were observed in the field, corrected by temperature and pressure for propagation error. Unfortunately, meteorological data are generally acquired only at the end points of a measured line. Using the mean of these meteorological values only approximates the actual conditions of the entire measured line and does not completely correct for errors in the propagation velocity of electromagnetic radiation. By applying the elevation, sea level, and scale factor corrections they were reduced to UTM grid distances [Ref. 4, pp. 124-125]. The UTM grid distances and standard deviations of the distances were determined in meters (Table III).

The angles were observed by using a Wild T-2 theodolite. To ensure the correctness of the beginning and ending azimuths, a check azimuth to a second station of known position was observed. The angles were observed at stations Moss 2, Mossback, Dune Temp, and Holm (Table IV). All observations were made by NPS students and conform to specification for a third-order class I traverse.

TABLE III

Grid Distances and Standard Deviations

Grid distances between stations			Distances (m.)	Standard deviation (m.)
Moss 2	and	Mossback	1,424.004	0.001
Mossback	and	Dune Temp	365.744	0.001
Dune Temp	and	Holm	6,476.271	0.003

TABLE IV

The Observed Angles and Standard Deviations

Backsight station	Center station	Foresight station	Observed angles °	'	"	Standard deviation "
Pipher	Moss 2	Mossback	246	05	43.200	01.984
Moss 2	Mossback	Dune Temp	222	51	08.600	01.405
Mossback	Dune Temp	Holm	190	15	02.600	01.203
Dune Temp	Holm	Moran	277	05	17.000	01.614

III. TRAVERSE COMPUTATIONS AND ADJUSTMENTS

A. INITIAL TRAVERSE COMPUTATIONS

A traverse (Figure 3.1) originates at station 1 (known position) and terminates at station 4 (also known position) (Table V). To compute the forward azimuth (Table VI) of an unknown leg, the angle is added to the back azimuth of the previous leg (Equation 3.1).

$$Az_i = Az_F + \sum_{j=1}^{i} \alpha_j - (i-1)\,180^\circ \qquad (3.1)$$

Where: i = the number of legs,

Az_i = the forward azimuth of an unknown leg, and

Az_F = the fixed initial azimuth.

For the computation of UTM grid coordinates, let $\triangle E_i$ and $\triangle N_i$ be designated as the departure and latitude for leg i (Figure 3.1) so that the general formulas to compute the departure and latitude are

$$\triangle E_i = d_i \sin Az_i \qquad (3.2)$$

$$\triangle N_i = d_i \cos Az_i \qquad (3.3)$$

where $i = 1, 2, 3, \ldots n$,

n = the number of the observed angles.

The algebraic signs of the departure and latitude for a traverse leg depend on the signs of the sine and cosine of the azimuth of that leg. The algebraic sign of the departure and latitude is determined by the following rules:

1. UTM grid azimuths are refered to grid north.
2. For azimuths between 0° and 180°, the departure is plus; for all other azimuths, the departure is minus.
3. For azimuths between 90° and 270°, the latitude is minus; for all other azimuths, the latitude is plus.

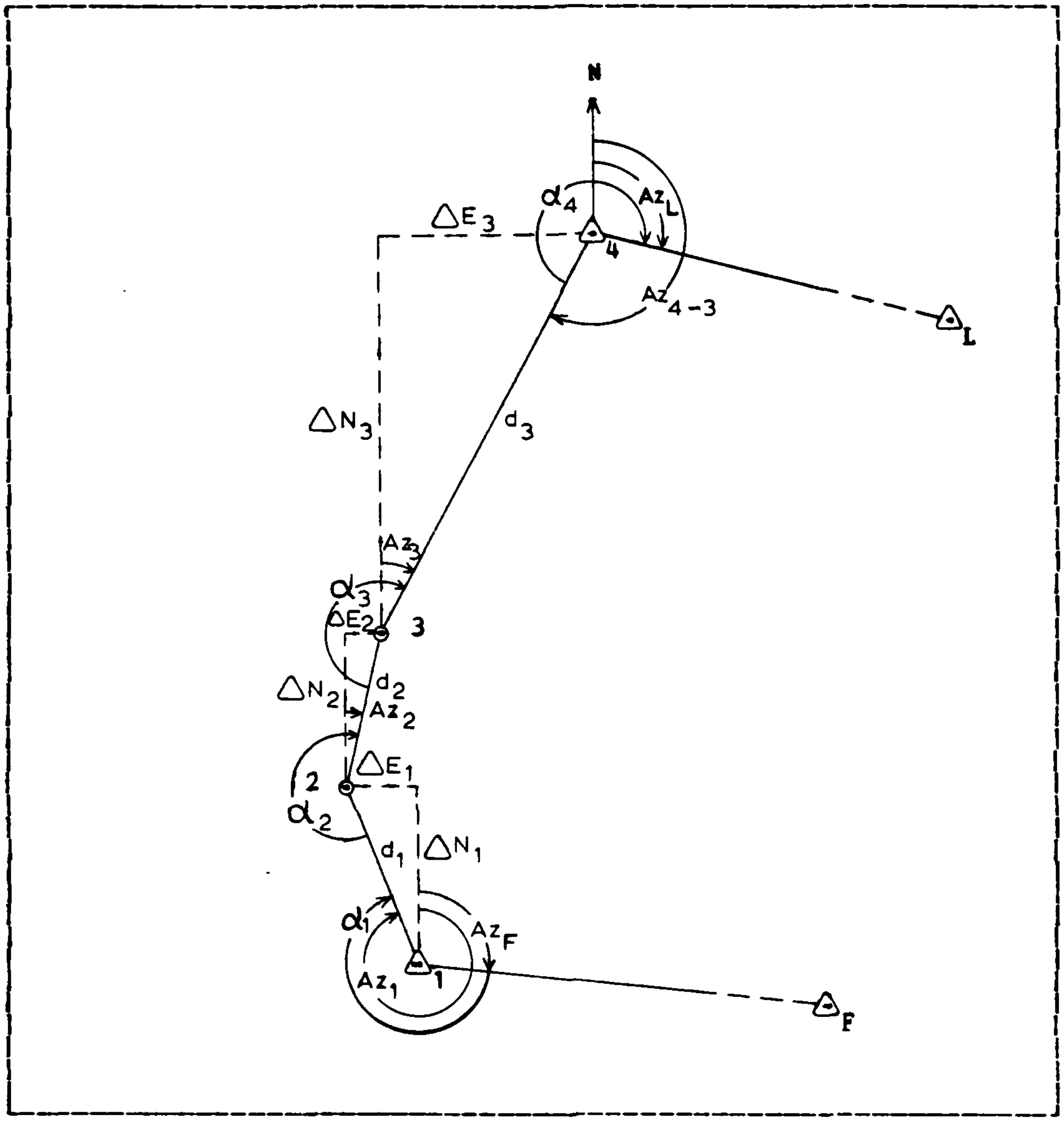


Figure 3.1 A Closed Traverse.

The calculators and computers with internal routines for trigonometric functions yield the proper sign automatically given the azimuth. The major portion of traverse computations consists of calculating departures and latitudes for successive legs and cumulating the values to determine the coordinates for consecutive traverse stations.

TABLE V

Data for Initial Traverse Computations

Leg	Distance (m.)	Angle
1 - 2	d_1 = 1,424.004	α_1 = 246° 05' 43.200"
2 - 3	d_2 = 365.744	α_2 = 222 51 08.600
3 - 4	d_3 = 6,476.271	α_3 = 190 15 02.600
The initial UTM grid azimuth		Az_F = 100 16 23.778
Station 1	UTM grid Northing	N_1 = 4,072,555.85206 m.
	UTM grid Easting	E_1 = 608,279.04404 m.

The coordinates of station j (j = i + 1) are E_j and N_j, the coordinates of station j are

$$E_j = E_i + \triangle E_i \tag{3.4}$$

$$N_j = N_i + \triangle N_i \tag{3.5}$$

where i = 1 , 2 , 3 , . . . n.

The values of departures, latitudes, and coordinates are shown in Table VII and were computed from the data in Tables V and VI by application of Equations (3.2) through (3.5).

B. COMPUTATION OF DISCREPANCIES

A closed traverse at Moss Landing (Figure 3.1) originated at station 1 (Moss 2) and closed at station 4 (Holm). The closing errors for a traverse are caused by observation errors in the observed angles and the measured distances. The closing errors may be computed by applying Equations (3.1) through (3.6).

TABLE VI

The Azimuth Calculation

Forward Azimuth	Back Azimuth	Observed Angle		°	'	"
	Az_F			100	16	23.778
		α_1		246	05	43.200
Az_1				346	22	06.978
			-	180	00	00.000
	Az_{2-1}			166	22	06.978
		α_2		222	51	08.600
Az_2				389	13	15.578
			-	180	00	00.000
	Az_{3-2}			209	13	15.578
		α_3		190	15	02.600
Az_3				399	28	18.178
			-	180	00	00.000
	Az_{4-3}			219	28	18.178

The angular error of closure may be computed by applying Equation (3.1) becomes

$$Az_F + \sum_{i=1}^{n} \alpha_i - (n-1)\,180^\circ - Az_L = W_1 \tag{3.6}$$

where Az_L is the fixed closing azimuth and W_1 is the angular error of closure. Az_L is the grid azimuth from stations Holm to Moran (Az_L = 136° 33' 26.334") (Table II). The computed closing azimuth is equal to Az_{4-3} plus the observed angle at station Holm = (219° 28' 18.178") + (277° 05' 17") = 496° 33' 35.178" = 136° 33' 35.178". From Equation (3.6), the angular error of closure is (136° 33' 35.178") - (136° 33' 26.334") = + 08.844".

TABLE VII

Initial Traverse Computations

		Values in meters			Values in meters
	N_1	4,072,555.85206		E_1	608,279.04404
$\triangle N_1$		1,383.89267	$\triangle E_1$		- 335.60166
	N_2	4,073,939.74473		E_2	607,943.44238
$\triangle N_2$		319.20061	$\triangle E_2$		178.54871
	N_3	4,074,258.94534		E_3	608,121.99109
$\triangle N_3$		4,999.28284	$\triangle E_3$		4,116.94755
	N_4	4,079,258.22818		E_4	612,238.93864

To compute the errors in closure in position for a closed traverse. Equations (3.2) to (3.5) are applied to obtain:

$$W_2 = \sum_{i=1}^{n-1} \triangle E_i - (E_T - E_1) \tag{3.7}$$

$$W_3 = \sum_{i=1}^{n-1} \triangle N_i - (N_T - N_1) \tag{3.8}$$

or

$$W_2 = E_n - E_T \tag{3.7a}$$

$$W_3 = N_n - N_T \tag{3.8a}$$

Where: n = the number of observed angles,

W_2 , W_3 = precalculated discrepancies,

E_1 , N_1 = the fixed initial coordinates,

E_T , N_T = the fixed closing coordinates, and

E_n , N_n = the computed closing coordinates.

The result of the errors in closure in position of a traverse is

$$\Delta d = (W_2^2 + W_3^2)^{1/2} \tag{3.9}$$

The computed closing coordinates of this traverse are E_4 = 612,238.93864 m and N_4 = 4,079,258.22818 m. The fixed closing coordinates are E_T = 612,238.85256 m and N_T = 4,079,258.31754 m. The precalculated discrepancies can be computed by using Equations (3.7a) and (3.8a). The results are W_2 = 0.08608 m and W_3 = -0.08936 m. By applying Equation (3.9), Δd is 0.12408 m. From Table III, the total measured distance is 8,266.019 m. The ratio of the distance error of closure to the total distance is 0.12408/8,266.019 or 1 part in 66,618. The ratio is an indication of the goodness of this traverse.

C. ADJUSTMENT OF A TRAVERSE BY AN APPROXIMATE METHOD

The angular error of closure of traverse may be distributed equally among the observed angles. The angular error of closure of this traverse is +8.844", which corresponds to a correction for each angle of -2.211". The observed angles were corrected by this value. The adjusted azimuth was then computed. The departures, latitudes, and coordinates were recomputed by using the adjusted azimuth (Table VIII). The closure corrections in E and N coordinates are +0.09635 m and -0.04326 m, respectively. The resultant closure is 0.10562 m. The ratio of the distance error of closure to the total distance is 0.10562/8,266.019 or 1 part in 78,262.

The adjustments of a closed traverse by the Approximate Method are completed by the compass rule which proportions the errors in E and N coordinates according to the distance of the course [Ref. 4, p. 354]. The corrections are applied to the departures and latitudes prior to computation of coordinates.

TABLE VIII

Initial Traverse Computations (Approximate Method with Adjusted Azimuth)

	Observed angle	Correction			Adjusted azimuth °	'	"
			Az_F		100	16	23.778
α_1	246° 05' 43.2"	- 2.211"			246	05	40.989
			Az_1		346	22	04.767
				-	180	00	00.000
			Az_{2-1}		166	22	04.767
α_2	222° 51' 08.6"	- 2.211"			222	51	06.389
			Az_2		389	13	11.156
				-	180	00	00.000
			Az_{3-2}		209	13	11.156
α_3	190° 15' 02.6"	- 2.211"			190	15	00.389
			Az_3		399	28	11.545
				-	180	00	00.000
			Az_{4-3}		219	28	11.545
α_4	277° 05' 17.0"	- 2.211"			277	05	14.789
			Az_{4-L}		136	33	26.334

		Values in meters			Values in meters
	N_1	4,072,555.85206		E_1	608,279.04404
$\triangle N_1$		1,383.88907	$\triangle E_1$		- 335.61649
	N_2	4,073,939.74113		E_2	607,943.42755
$\triangle N_2$		319.20444	$\triangle E_2$		178.54187
	N_3	4,074,258.94557		E_3	608,121.96942
$\triangle N_3$		4,999.41523	$\triangle E_3$		4,116.78679
	N_4	4,079,258.36080		E_4	612,238.75621
	N_T	4,079,258.31754		E_T	612,238.85256
	dN	0.04326		dE	- 0.09635

$$\delta E_i = (dE / D) \cdot d_i \tag{3.10}$$

$$\delta N_i = (dN / D) \cdot d_i \tag{3.11}$$

Where: δE_i = correction to $\triangle E_i$,

δN_i = correction to $\triangle N_i$,

dE = total closure correction in the E coordinate,

dN = total closure correction in the N coordinate, and

D = total distance.

The adjusted E and N coordinates (Table IX) were different from the fixed E and N coordinates (Table II) by 0.00001 m due to round off error. The calculations and adjustments were illustrated in sections A, B, and C by using the hand calculator (TI-59) and rounded off at 5 decimal places. The computer program was written to calculate the adjusted traverse station positions by the Approximate Method by using 16 decimal places (Appendix A). The approximate traverse adjustment is based on an assumed condition that the angular precision equals the precision in linear distance.

D. ADJUSTMENT OF A TRAVERSE BY LEAST SQUARES METHODS

The method of least squares provides a rigorous adjustment and best estimates for positions of all traverse stations. The Least Squares Method is used to simultaneously eliminate closing errors in azimuths and coordinates of traverses.

1. The Principle of Least Squares

The fundamental condition of the least squares technique in surveying requires that the sum of the squares of the residuals be minimized. A residual is defined as the difference between the true and observed values. In making

TABLE IX

Adjusted Coordinates (Approximate Method)

	Departure	Correction	Grid	Values in meters
			E_1	608,279.04404
$\triangle E_1$	-335.61649	+ 0.01659		- 335.59990
			E_2	607,943.44414
$\triangle E_2$	178.54187	+ 0.00426		178.54613
			E_3	608,121.99027
$\triangle E_3$	4,116.78679	+ 0.07549		4,116.86228
			E_4	612,238.85255

	Latitude	Correction	Grid	Values in meters
			N_1	4,072,555.85206
$\triangle N_1$	1,383.88907	- 0.00745		1,383.88162
			N_2	4,073,939.73368
$\triangle N_2$	319.20444	- 0.00191		319.20253
			N_3	4,074,258.93621
$\triangle N_3$	4,999.41523	- 0.03389		4,999.38134
			N_4	4,079,258.31755

physical measurements, the true values can never be determined. The least squares principle establishes a criterion for obtaining the best estimates of the true values.

If the best estimates of the true values are stated by x_i and observed values by $\bar{x}_i$, the residuals are expressed as

$$v_i = x_i - \bar{x}_i$$

The fundamental condition of least squares, for uncorrelated observations with equal precision are expressed as

$$\sum_{i=1}^{n} (v_i)^2 = (v_1)^2 + (v_2)^2 + (v_3)^2 + \ldots + (v_n)^2 = \text{Minimum}$$

or in matrix form

$$V^T V = \text{Minimum} \qquad (3.12)$$

In general, the observed values are of unequal precision. The observed value of high precision has a small variance. Conversely, a low precision of the observed value has a large variance. Since the value of the variance goes in opposite direction to that of the precision, the observation is assigned a value called weight corresponding to a quality that is inversely proportional to the observation's variance.

For uncorrelated measurements x_1, x_2, x_3, . . . x_n, with variances σ_1^2, σ_2^2, σ_3^2, . . . σ_n^2, respectively, the weights of these uncorrelated measurements are,

$$p_{11} = \sigma_\circ^2 / \sigma_1^2,\ p_{22} = \sigma_\circ^2 / \sigma_2^2,\ p_{33} = \sigma_\circ^2 / \sigma_3^2,\ \ldots .$$

$$p_{nn} = \sigma_\circ^2 / \sigma_n^2 \qquad (3.13)$$

where $\sigma_\circ^2$ is the proportionally constant of an observation of unit weight [Ref. 7, p. 67]. These weights may be collected into a corresponding diagonal P matrix, called the weight matrix:

$$P = \begin{bmatrix} p_{11} & 0 & 0 & . & . & 0 \\ 0 & p_{22} & 0 & . & . & 0 \\ 0 & 0 & p_{33} & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & p_{nn} \end{bmatrix}$$

The weight matrix consists of weighted observations of a traverse with mixed kinds of measurements. They are distances and angles. The variance of a measured distance is expressed in meters and an angle in radians. For uncorrelated observations of unequal precision, the fundamental condition of least squares is expressed as

$$\sum_{i=1}^{n} p_{ii} v_i^2 = p_{11} v_1^2 + p_{22} v_2^2 + p_{33} v_3^2 + \cdots + p_{nn} v_n^2$$
$$= \text{Minimum}$$

or in matrix form

$$V^T P V = \text{Minimum} \tag{3.14}$$

In the problems involving the adjustment of observed values, all observed quantities are expressed by functions of the quantities to be determined. In simple cases these relations are linear, but when this is not the case, the relations must be converted into the linear form by expanding them into Taylor's series. The terms of higher order are neglected, so as to obtain linear relations, solving the resulting linear equations, then iterating until the effect of the neglected higher order terms are minimized.

2. Least Squares Adjustment of Indirect Observations

The observed values are related to the desired unknown values through formulas or functions which are called observation equations. One observation equation is written for each measurement. To solve for the best value of each unknown parameter, at least one redundant observation equation must be written. That is, the number observations must be greater than the number of unknowns. The linear observation equations can be written in the general form as follows:

$$
\begin{array}{cccccccc}
a_{10} + & a_{11}x_1 & + a_{12}x_2 & + \cdots & + a_{1m}x_m & = G_1 & + v_1 \\
a_{20} + & a_{21}x_1 & + a_{22}x_2 & + \cdots & + a_{2m}x_m & = G_2 & + v_2 \\
\cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\
a_{n0} + & a_{n1}x_1 & + a_{n2}x_2 & + \cdots & + a_{nm}x_m & = G_n & + v_n
\end{array}
$$

where n is the number of observations; m is the number of the unknowns; a_{10}, a_{20}, . . . a_{n0} are constants; a_{11}, a_{12}, . . . a_{nm} are coefficients of the unknowns x_1, x_2, . . . x_m; and v_1, v_2, . . . v_n are the residuals.

Because the observations G_i (i = 1, 2, . . . n) are not free from random errors, each G_i must be corrected by a residual value, v_i. Let $b_i = G_i - a_{i0}$. Thus

$$
\begin{array}{cccccc}
a_{11}x_1 & + a_{12}x_2 & + \cdots + & a_{1m}x_m & - b_1 & = v_1 \\
a_{21}x_1 & + a_{22}x_2 & + \cdots + & a_{2m}x_m & - b_2 & = v_2 \\
\cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & & \cdot & \cdot & \cdot \\
\cdot & \cdot & & \cdot & \cdot & \cdot \\
a_{n1}x_1 & + a_{n2}x_2 & + \cdots + & a_{nm}x_m & - b_n & = v_n
\end{array}
$$

or in the matrix form

$$V = AX - B \qquad (3.15)$$

This equation is called the observation equation or the observation equation matrix.

For uncorrelated observations of unequal precision, substitute the value for the V matrix from the observation Equation (3.15) into Equation (3.14)

$$
\begin{aligned}
V^T P V &= (AX - B)^T P (AX - B) \\
&= [(AX)^T - B^T] P (AX - B) \\
&= (X^T A^T - B^T) P (AX - B) \\
&= (X^T A^T P - B^T P)(AX - B) \\
&= X^T A^T PAX - X^T A^T PB - B^T PAX + B^T PB \\
&= X^T A^T PAX - X^T A^T PB - X^T A^T PB + B^T PB \\
&= X^T A^T PAX - 2X^T A^T PB + B^T PB \qquad (3.16)
\end{aligned}
$$

[from matrix algebra, $(AX)^T = X^T A^T$ and $B^T PAX = X^T A^T PB$]

The minimum of this function can be found by taking the partial derivatives of the function with respect to each unknown variable (i.e., with respect to the $x_1, x_2, \ldots x_m$) must equal zero. Hence,

$$\frac{\partial}{\partial x}(V^T PV) = 2A^T PAX - 2A^T PB = 0 \qquad (3.17)$$

Dividing Equation (3.17) by 2, the following result is obtained:

$$A^T PAX = A^T PB$$

This represents the normal equations, and by multiplying this equation by $(A^T PA)^{-1}$, the solution is obtained:

$$X = (A^T PA)^{-1} A^T PB \qquad (3.18)$$

For uncorrelated observations with equal precision, the weight matrix is an identity matrix, and equation (3.18) becomes

$$X = (A^T A)^{-1} A^T B \qquad (3.19)$$

This equation can be derived similarly to the unequal precision case. Equations (3.18) and (3.19) are the basic least squares matrix equations.

If the relations are nonlinear, the relations must be converted into the linear form by using Taylor's series expansion [Ref. 3, p. 919]. Let $F = f(x_1, x_2, \ldots x_m)$ be the general observation equation that is a nonlinear function. The Taylor's series expansion is

$$F = f(x_1^\circ, x_2^\circ, \ldots x_m^\circ) + \frac{\partial f}{\partial x_1^\circ}\delta x_1 + \frac{\partial f}{\partial x_2^\circ}\delta x_2 + \ldots + \frac{\partial f}{\partial x_m^\circ}\delta x_m$$
$$+ \text{ Higher order terms}$$

where $x_1^\circ, x_2^\circ, \ldots x_m^\circ$ are the approximate values of the variables at which the function is evaluated. For the traverse problems, an approximate value can be the precalculated value, and $\delta x_1, \delta x_2, \ldots \delta x_m$ can be the corrections. The higher order terms in the series are neglected and only the zero and first order terms are maintained, the approximate value must be improved by successive iterations until the effect of the neglected higher order terms is minimized. After linearization, the observation equations become:

$$v_1 = f_1(x_1^\circ, x_2^\circ, \ldots x_m^\circ) + \frac{\partial f_1}{\partial x_1^\circ}\delta x_1 + \frac{\partial f_1}{\partial x_2^\circ}\delta x_2 + \ldots + \frac{\partial f_1}{\partial x_m^\circ}\delta x_m - G_1$$

$$v_2 = f_2(x_1^\circ, x_2^\circ, \ldots x_m^\circ) + \frac{\partial f_2}{\partial x_1^\circ}\delta x_1 + \frac{\partial f_2}{\partial x_2^\circ}\delta x_2 + \ldots + \frac{\partial f_2}{\partial x_m^\circ}\delta x_m - G_2$$

$$\vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots$$

$$v_n = f_n(x_1^\circ, x_2^\circ, \ldots x_m^\circ) + \frac{\partial f_n}{\partial x_1^\circ}\delta x_1 + \frac{\partial f_n}{\partial x_2^\circ}\delta x_2 + \ldots + \frac{\partial f_n}{\partial x_m^\circ}\delta x_m - G_n$$

or in the matrix form

$$V = AX - B \tag{3.20}$$

where

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1^\circ} & \frac{\partial f_1}{\partial x_2^\circ} & \cdots\cdots & \frac{\partial f_1}{\partial x_m^\circ} \\ \frac{\partial f_2}{\partial x_1^\circ} & \frac{\partial f_2}{\partial x_2^\circ} & \cdots\cdots & \frac{\partial f_2}{\partial x_m^\circ} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial f_n}{\partial x_1^\circ} & \frac{\partial f_n}{\partial x_2^\circ} & \cdots\cdots & \frac{\partial f_n}{\partial x_m^\circ} \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_n \end{bmatrix}$$

$$B = \begin{bmatrix} G_1 - f_1(x_1^\circ, x_2^\circ, \ldots\ldots x_m^\circ) \\ G_2 - f_2(x_1^\circ, x_2^\circ, \ldots\ldots x_m^\circ) \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad\quad \cdot \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad\quad \cdot \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad\quad \cdot \\ G_n - f_n(x_1^\circ, x_2^\circ, \ldots\ldots x_m^\circ) \end{bmatrix} \qquad X = \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta x_m \end{bmatrix}$$

The remainder of the least squares procedure is the same as indicated by Equations (3.16) to (3.19).

There are two types of observation equations for the adjustment of the traverse. They are the angle and distance observation equations. Both of them are nonlinear, and they must be linearized using the Taylor's series.

To derive the observation equation for the angle α_i (Figure 3.1), first write the two azimuth observation equations for the azimuth Az_{ik} and Az_{ij}, where Az_{ik} is the back azimuth and Az_{ij} is the forward azimuth.

$$Az_{ik} = \arctan \frac{E_k - E_i}{N_k - N_i}$$

$$Az_{ij} = \arctan \frac{E_j - E_i}{N_j - N_i}$$

Then, the angle observation equation for the angle α_i is

$$v_i = Az_{ij} - Az_{ik} - \alpha_i$$
$$= \text{arc tan}\,\frac{E_j - E_i}{N_j - N_i} - \text{arc tan}\,\frac{E_k - E_i}{N_k - N_i} - \alpha_i \qquad (3.21)$$

Equation (3.21) is nonlinear in the parameters. Write this equation in function form:

$$v_i = f_i\,(E_i,\ N_i,\ E_j,\ N_j,\ E_k,\ N_k\,) - \alpha_i$$

Thus, the linearized form of the angle observation equation is

$$v_i = f_i\,(E_i^\circ,\ N_i^\circ,\ E_j^\circ,\ N_j^\circ,\ E_k^\circ,\ N_k^\circ\,) + \frac{\partial f_i}{\partial E_i^\circ}\delta E_i + \frac{\partial f_i}{\partial N_i^\circ}\delta N_i$$
$$+ \frac{\partial f_i}{\partial E_j^\circ}\delta E_j + \frac{\partial f_i}{\partial N_j^\circ}\delta N_j + \frac{\partial f_i}{\partial E_k^\circ}\delta E_k + \frac{\partial f_i}{\partial N_k^\circ}\delta N_k - \alpha_i$$

The observation equation for the distance d_i between two points i and j is

$$v_i = [\,(E_i - E_j)^2 + (N_i - N_j)^2]^{1/2} - d_i$$

or in the functional form

$$v_i = f_i\,(E_i,\ N_i,\ E_j,\ N_j\,) - d_i$$

Thus, the linearized form of the distance observation equation is

$$v_i = f_i\,(E_i^\circ,\ N_i^\circ,\ E_j^\circ,\ N_j^\circ\,) + \frac{\partial f_i}{\partial E_i^\circ}\delta E_i + \frac{\partial f_i}{\partial N_i^\circ}\delta N_i$$
$$+ \frac{\partial f_i}{\partial E_j^\circ}\delta E_j + \frac{\partial f_i}{\partial N_j^\circ}\delta N_j - d_i$$

For the adjustment of the traverse which was conducted at Moss Landing with the technique of adjustment of indirect observations, four angle and three distance

observation equations need to be written. The seven equations would include as unknown parameters the four coordinates of stations 2 (Mossback) and 3 (Dune Temp). Four normal equations are formed and solved for corrections to the approximate values (precalculated values) of the parameters. The corrections are added to the approximations to update their values, and the solution is repeated until the last set of corrections is insignificantly small.

The linearized equations of this problems are

$$
\begin{aligned}
v_1 &= a_{11}\delta E_2 + a_{12}\delta N_2 + a_{13}\delta E_3 + a_{14}\delta N_3 - b_1 \\
v_2 &= a_{21}\delta E_2 + a_{22}\delta N_2 + a_{23}\delta E_3 + a_{24}\delta N_3 - b_2 \\
&\vdots \\
v_7 &= a_{71}\delta E_2 + a_{72}\delta N_2 + a_{73}\delta E_3 + a_{74}\delta N_3 - b_7
\end{aligned}
$$

The zero order terms of the angle and distance functions are

$$
\begin{aligned}
f_1 &= \arctan \frac{E_2^\circ - E_1}{N_2^\circ - N_1} - Az_F \\
f_2 &= \arctan \frac{E_3^\circ - E_2^\circ}{N_3^\circ - N_2^\circ} - \arctan \frac{E_1 - E_2^\circ}{N_1 - N_2^\circ} \\
f_3 &= \arctan \frac{E_4 - E_3^\circ}{N_4 - N_3^\circ} - \arctan \frac{E_2^\circ - E_3^\circ}{N_2^\circ - N_3^\circ} \\
f_4 &= Az_L - \arctan \frac{E_3^\circ - E_4}{N_3^\circ - N_4} \\
f_5 &= [(E_1 - E_2^\circ)^2 + (N_1 - N_2^\circ)^2]^{1/2} \\
f_6 &= [(E_2^\circ - E_3^\circ)^2 + (N_2^\circ - N_3^\circ)^2]^{1/2} \\
f_7 &= [(E_3^\circ - E_4)^2 + (N_3^\circ - N_4)^2]^{1/2}
\end{aligned}
$$

By using the data of the known coordinates (Table II) and the approximate values (Table VII)

$$
a_{11} = \frac{\partial f_1}{\partial E_2^\circ} = \left[\frac{N_2^\circ - N_1}{(N_2^\circ - N_1)^2 + (E_2^\circ - E_1)^2}\right] = 0.68246 \times 10^{-3}
$$

$$a_{12} = \frac{\partial f_1}{\partial N_2^\circ} = -\left[\frac{E_2^\circ - E_1}{(N_2^\circ - N_1)^2 + (E_2^\circ - E_1)^2}\right] = 0.16550 \times 10^{-3}$$

$$a_{13} = \frac{\partial f_1}{\partial E_3^\circ} = 0$$

$$a_{14} = \frac{\partial f_1}{\partial N_3^\circ} = 0$$

$$a_{21} = \frac{\partial f_2}{\partial E_2^\circ} = -\left[\frac{N_3^\circ - N_2^\circ}{(N_3^\circ - N_2^\circ)^2 + (E_3^\circ - E_2^\circ)^2}\right] + \left[\frac{N_1 - N_2^\circ}{(N_1 - N_2^\circ)^2 + (E_1 - E_2^\circ)^2}\right]$$

$$= -3.06868 \times 10^{-3}$$

$$a_{22} = \frac{\partial f_2}{\partial N_2^\circ} = \left[\frac{E_3^\circ - E_2^\circ}{(N_3^\circ - N_2^\circ)^2 + (E_3^\circ - E_2^\circ)^2}\right] - \left[\frac{E_1 - E_2^\circ}{(N_1 - N_2^\circ)^2 + (E_1 - E_2^\circ)^2}\right]$$

$$= 1.16926 \times 10^{-3}$$

$$a_{23} = \frac{\partial f_2}{\partial E_3^\circ} = \left[\frac{N_3^\circ - N_2^\circ}{(N_3^\circ - N_2^\circ)^2 + (E_3^\circ - E_2^\circ)^2}\right] = 2.38621 \times 10^{-3}$$

$$a_{24} = \frac{\partial f_2}{\partial N_3^\circ} = -\left[\frac{E_3^\circ - E_2^\circ}{(N_3^\circ - N_2^\circ)^2 + (E_3^\circ - E_2^\circ)^2}\right] = -1.33476 \times 10^{-3}$$

$$a_{31} = \frac{\partial f_3}{\partial E_2^\circ} = -\left[\frac{N_2^\circ - N_3^\circ}{(N_2^\circ - N_3^\circ)^2 + (E_2^\circ - E_3^\circ)^2}\right] = 2.38621 \times 10^{-3}$$

$$a_{32} = \frac{\partial f_3}{\partial N_2^\circ} = \left[\frac{E_2^\circ - E_3^\circ}{(N_2^\circ - N_3^\circ)^2 + (E_2^\circ - E_3^\circ)^2}\right] = -1.33476 \times 10^{-3}$$

$$a_{33} = \frac{\partial f_3}{\partial E_3^\circ} = -\left[\frac{N_4 - N_3^\circ}{(N_4 - N_3^\circ)^2 + (E_4 - E_3^\circ)^2}\right] + \left[\frac{N_2^\circ - N_3^\circ}{(N_2^\circ - N_3^\circ)^2 + (E_2^\circ - E_3^\circ)^2}\right]$$

$$= -2.50541 \times 10^{-3}$$

$$a_{34} = \frac{\partial f_3}{\partial N_3^\circ} = \left[\frac{E_4 - E_3^\circ}{(N_4 - N_3^\circ)^2 + (E_4 - E_3^\circ)^2}\right] - \left[\frac{E_2^\circ - E_3^\circ}{(N_2^\circ - N_3^\circ)^2 + (E_2^\circ - E_3^\circ)^2}\right]$$

$$= 1.43291 \times 10^{-3}$$

$$a_{41} = \frac{\partial f_4}{\partial E_2^\circ} = 0$$

$$a_{42} = \frac{\partial f_4}{\partial N_2^\circ} = 0$$

$$a_{43} = \frac{\partial f_4}{\partial E_3^\circ} = -\left[\frac{N_3^\circ - N_4}{(N_3^\circ - N_4)^2 + (E_3^\circ - E_4)^2}\right] = 0.11920 \times 10^{-3}$$

$$a_{44} = \frac{\partial f_4}{\partial N_3^\circ} = \left[\frac{E_3^\circ - E_4}{(N_3^\circ - N_4)^2 + (E_3^\circ - E_4)^2}\right] = -0.09816 \times 10^{-3}$$

$$a_{51} = \frac{\partial f_5}{\partial E_2^\circ} = -\left[\frac{E_1 - E_2^\circ}{\left[(E_1 - E_2^\circ)^2 + (N_1 - N_2^\circ)^2\right]^{1/2}}\right] = -235.67466 \times 10^{-3}$$

$$a_{52} = \frac{\partial f_5}{\partial N_2^\circ} = -\left[\frac{N_1 - N_2^\circ}{\left[(E_1 - E_2^\circ)^2 + (N_1 - N_2^\circ)^2\right]^{1/2}}\right] = 971.83201 \times 10^{-3}$$

$$a_{53} = \frac{\partial f_5}{\partial E_3^\circ} = 0$$

$$a_{54} = \frac{\partial f_5}{\partial N_3^\circ} = 0$$

$$a_{61} = \frac{\partial f_6}{\partial E_2^\circ} = \left[\frac{E_2^\circ - E_3^\circ}{\left[(E_2^\circ - E_3^\circ)^2 + (N_2^\circ - N_3^\circ)^2\right]^{1/2}}\right] = -488.17948 \times 10^{-3}$$

$$a_{62} = \frac{\partial f_6}{\partial N_2^\circ} = \left[\frac{N_2^\circ - N_3^\circ}{\left[(E_2^\circ - E_3^\circ)^2 + (N_2^\circ - N_3^\circ)^2\right]^{1/2}}\right] = -872.74326 \times 10^{-3}$$

$$a_{63} = \frac{\partial f_6}{\partial E_3^\circ} = -\left[\frac{E_2^\circ - E_3^\circ}{\left[(E_2^\circ - E_3^\circ)^2 + (N_2^\circ - N_3^\circ)^2\right]^{1/2}}\right] = 488.17948 \times 10^{-3}$$

$$a_{64} = \frac{\partial f_6}{\partial N_3^\circ} = -\left[\frac{N_2^\circ - N_3^\circ}{\left[(E_2^\circ - E_3^\circ)^2 + (N_2^\circ - N_3^\circ)^2\right]^{1/2}}\right] = 872.74326 \times 10^{-3}$$

$$a_{71} = \frac{\partial f_7}{\partial E_2^\circ} = 0$$

$$a_{72} = \frac{\partial f_7}{\partial N_2^\circ} = 0$$

$$a_{73} = \frac{\partial f_7}{\partial E_3^\circ} = \left[\frac{E_3^\circ - E_4}{\left[(E_3^\circ - E_4)^2 + (N_3^\circ - N_4)^2\right]^{1/2}}\right] = -\ 635.68254 \times 10^{-3}$$

$$a_{74} = \frac{\partial f_7}{\partial N_3^\circ} = \left[\frac{N_3^\circ - N_4}{\left[(E_3^\circ - E_4)^2 + (N_3^\circ - N_4)^2\right]^{1/2}}\right] = -\ 771.95059 \times 10^{-3}$$

or in the matrix form

$$A = 10^{-3}\begin{bmatrix} 0.68246 & 0.16550 & 0.00000 & 0.00000 \\ -3.06868 & 1.16926 & 2.38621 & -1.33476 \\ 2.38621 & -1.33476 & -2.50541 & 1.43291 \\ 0.00000 & 0.00000 & 0.11920 & -0.09816 \\ -235.67466 & 971.83201 & 0.00000 & 0.00000 \\ -448.17948 & -872.74326 & 488.17948 & 872.74326 \\ 0.00000 & 0.00000 & -635.68254 & -771.95059 \end{bmatrix}$$

The computation of B matrix, by using the data from Tables II, III, IV, and VII

$$\begin{aligned} b_1 &= \alpha_1 - f_1 = 0.00000 \text{ radians} \\ b_2 &= \alpha_2 - f_2 = 0.00000 \text{ radians} \\ b_3 &= \alpha_3 - f_3 = 0.00002 \text{ radians} \\ b_4 &= \alpha_4 - f_4 = 0.00002 \text{ radians} \\ b_5 &= d_1 - f_5 = 0.00000 \text{ meters} \\ b_6 &= d_2 - f_6 = 0.00000 \text{ meters} \\ b_7 &= d_3 - f_7 = -0.01426 \text{ meters} \end{aligned}$$

or in the matrix form

$$B^T = [\ 0 \quad 0 \quad 0.00002 \quad 0.00002 \quad 0 \quad 0 \quad -0.01426\ ]$$

The weight matrix can be computed by using the variance of the observed angles and distances from Tables III and IV by applying to Equation (3.13). For uncorrelated observations of unequal precision, the weight matrix is defined as the diagonal P matrix.

$$
\begin{aligned}
p_{11} &= 1 / \sin^2(1.984'') &&= 10{,}808{,}537{,}426.79957 \\
p_{22} &= 1 / \sin^2(1.405'') &&= 21{,}552{,}498{,}219.02474 \\
p_{33} &= 1 / \sin^2(1.203'') &&= 29{,}398{,}082{,}997.43487 \\
p_{44} &= 1 / \sin^2(1.614'') &&= 16{,}332{,}144{,}194.08730 \\
p_{55} &= 1 / (0.001)^2 &&= 1{,}000{,}000.00000 \\
p_{66} &= 1 / (0.001)^2 &&= 1{,}000{,}000.00000 \\
p_{77} &= 1 / (0.003)^2 &&= 111{,}111.11111
\end{aligned}
$$

For uncorrelated observations of unequal precision, the values of A, B, and P matrices are applied to Equation (3.18). The correction vector is found to be

$$
X = \begin{bmatrix} \delta E_2 = 0.01284 \\ \delta N_2 = 0.00336 \\ \delta E_3 = 0.01079 \\ \delta N_3 = 0.00495 \end{bmatrix}
$$

The approximation values were improved by adding the corrections to the first approximation values. The improved approximation values after the first iteration are

$$
\begin{aligned}
E_2 &= 607{,}943.44238 + 0.01284 = 607{,}943.45522 \text{ m.} \\
N_2 &= 4{,}073{,}939.74473 + 0.00336 = 4{,}073{,}939.74809 \text{ m.} \\
E_3 &= 608{,}121.99110 + 0.01079 = 608{,}122.00189 \text{ m.} \\
N_3 &= 4{,}074{,}258.94534 + 0.00495 = 4{,}074{,}258.95029 \text{ m.}
\end{aligned}
$$

Using these values, the solution is iterated. After the second iteration, the correction vector is zero to six decimal places, so the improved approximation values are the final estimates of the coordinates.

For uncorrelated observations of equal precision, the correction vector is found by solving Equation (3.19). The correction vector after the first iteration is

$$
X = \begin{bmatrix} \delta E_2 = 0.01756 \\ \delta N_2 = 0.00426 \\ \delta E_3 = 0.01660 \\ \delta N_3 = 0.00479 \end{bmatrix}
$$

The improved approximation values after the first iteration are

$$E_2 = 607{,}943.44238 + 0.01756 = 607{,}943.45994 \text{ m.}$$
$$N_2 = 4{,}073{,}939.74473 + 0.00426 = 4{,}073{,}939.74899 \text{ m.}$$
$$E_3 = 608{,}121.99110 + 0.01660 = 608{,}122.00770 \text{ m.}$$
$$N_3 = 4{,}074{,}258.94534 + 0.00479 = 4{,}074{,}258.95013 \text{ m.}$$

The solution is iterated by using these adjusted values. The correction vector is zero to six decimal places after the second iteration. These values are the final estimates of the coordinates.

The standard deviation of an observation which has unit weight can be found by the following equations, for uncorrelated observations of unequal precision,

$$\sigma_o = \left[\frac{V^T P V}{n-m}\right]^{1/2} \tag{3.22}$$

for uncorrelated observations of equal precision,

$$\sigma_o = \left[\frac{V^T V}{n-m}\right]^{1/2} \tag{3.23}$$

where n is the number of observation equations and m is the number of unknowns [Ref. 7, p. 249]. After calculating the best estimate values of the unknowns, the V matrix can be computed from Equation (3.20).

The standard deviations of the best estimate values for the unknowns are then given by the following equations:

$$\sigma_i = \sigma_o \left[S_{ii}\right]^{1/2} \tag{3.24}$$

where σ_i is the standard deviation of the ith adjusted quantity. The quantity in the ith row of the X matrix, S_{ii} is an element of the $(A^T PA)^{-1}$ matrix for uncorrelated observations of unequal precision. For uncorrelated observations of equal precision, S_{ii} is an element of the $(A^T A)^{-1}$ matrix [Ref. 7, p. 250].

$$(A^T PA)^{-1} \text{ or } (A^T A)^{-1} = \begin{bmatrix} S_{11} & S_{12} & \cdots\cdots & S_{1m} \\ S_{21} & S_{22} & \cdots\cdots & S_{2m} \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ S_{m1} & S_{m2} & \cdots\cdots & S_{mm} \end{bmatrix}$$

The standard deviations of adjusted angles and distances can be computed by the following equation:

$$\sigma_i = \sigma_o \left[Q_{ii} \right]^{1/2} \qquad (3.25)$$

where σ_i is the standard deviations of the ith adjusted quantity. The quantity in the ith row of the V matrix, Q_{ii} is an element of $A(A^T PA)^{-1}A^T$ matrix for uncorrelated observations of unequal precision. For uncorrelated observations of equal precision, Q_{ii} is an element of the $A(A^T A)^{-1}A^T$ matrix [Ref. 3, p. 912].

$$\begin{matrix} A(A^T PA)^{-1}A^T \\ \text{or} \\ A(A^T A)^{-1}A^T \end{matrix} = \begin{bmatrix} Q_{11} & Q_{12} & \cdots\cdots & Q_{1n} \\ Q_{21} & Q_{22} & \cdots\cdots & Q_{2n} \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdots\cdots & \cdot \\ Q_{n1} & Q_{n2} & \cdots\cdots & Q_{nn} \end{bmatrix}$$

A closed traverse was conducted at Moss Landing. After the second iteration, the V matrix can be found by applying the values of A, B, and X matrices to Equation (3.20).

For uncorrelated observations of unequal precision, the V matrix is found to be

$$V = \begin{bmatrix} v_{a1} = 0.93149 \times 10^{-5} \text{ radians or } 1.92134'' \\ v_{a2} = -1.63120 \times 10^{-5} \text{ radians or } -3.36459'' \\ v_{a3} = -1.28363 \times 10^{-5} \text{ radians or } -2.64768'' \\ v_{a4} = -2.30436 \times 10^{-5} \text{ radians or } -4.75308'' \\ v_{d1} = 0.00024 \text{ meters} \\ v_{d2} = 0.00039 \text{ meters} \\ v_{d3} = 0.00358 \text{ meters} \end{bmatrix}$$

The standard deviation of unit weight is found by solving Equation (3.22), the result is

$$\sigma_{\circ} = \pm\ 2.69685$$

The values of $(A^T PA)^{-1}$ are

$$10^{-5} \begin{bmatrix} 1.25178 & 0.22019 & 1.25575 & 0.10788 \\ 0.22019 & 0.13819 & 0.21810 & 0.11127 \\ 1.25575 & 0.21810 & 1.46408 & 0.03815 \\ 0.10788 & 0.11127 & 0.03815 & 0.22515 \end{bmatrix}$$

The standard deviations of the best estimate values for positions can be found by solving Equation (3.24), the results are

$$\begin{aligned} \sigma_{E2} &= \pm\ 0.00954 \text{ meters} \\ \sigma_{N2} &= \pm\ 0.00317 \text{ meters} \\ \sigma_{E3} &= \pm\ 0.01032 \text{ meters} \\ \sigma_{N3} &= \pm\ 0.00405 \text{ meters} \end{aligned}$$

The values of $A(A^T PA)^{-1}A^T$ are

$$10^{-10}\begin{bmatrix} 0.064 & -0.052 & -0.021 & 0.010 & -4.165 & -6.963 & -63.88 \\ -0.052 & 0.225 & -0.175 & 0.002 & 3.415 & 4.690 & 41.59 \\ -0.021 & -0.175 & 0.211 & -0.014 & 2.563 & 3.574 & 31.79 \\ 0.010 & 0.002 & -0.014 & 0.002 & -1.813 & -1.301 & -9.501 \\ -4.165 & 3.415 & 2.563 & -1.813 & 9917 & -117.4 & -1047 \\ -6.963 & 4.690 & 3.574 & -1.301 & -117.4 & 9830 & -1525 \\ -63.88 & 41.59 & 31.79 & -9.501 & -1047 & -1525 & 76323 \end{bmatrix}$$

The standard deviations of adjusted angles and distances are found by solving Equation (3.25), the results are

$$\begin{aligned} \sigma_{a1} &= \pm\ 1.40346 \text{ seconds} \\ \sigma_{a2} &= \pm\ 2.64103 \text{ seconds} \\ \sigma_{a3} &= \pm\ 2.55260 \text{ seconds} \\ \sigma_{a4} &= \pm\ 0.26137 \text{ seconds} \\ \sigma_{d1} &= \pm\ 0.00269 \text{ meters} \\ \sigma_{d2} &= \pm\ 0.00267 \text{ meters} \\ \sigma_{d3} &= \pm\ 0.00745 \text{ meters} \end{aligned}$$

For uncorrelated observations of equal precision, the V matrix is found to be

$$V = \begin{bmatrix} v_{a1} = 1.26896 \times 10^{-5} \text{ radians or } 2.61742'' \\ v_{a2} = -1.56893 \times 10^{-5} \text{ radians or } -3.23615'' \\ v_{a3} = -1.75414 \times 10^{-5} \text{ radians or } -3.61817'' \\ v_{a4} = -2.23358 \times 10^{-5} \text{ radians or } -4.60710'' \\ v_{d1} = 0.14037 \times 10^{-7} \text{ meters} \\ v_{d2} = 0.23844 \times 10^{-7} \text{ meters} \\ v_{d3} = 0.24365 \times 10^{-7} \text{ meters} \end{bmatrix}$$

The standard deviation of unit weight is found by solving Equation (3.23), the result is

$$\sigma_\circ = \pm\ 2.01144 \times 10^{-5}$$

The values of $(A^T A)^{-1}$ are

$$10^{-4}\begin{bmatrix} 0.28193 & 0.06827 & -0.85449 & 0.70380 \\ 0.06827 & 0.01664 & -0.20731 & 0.17075 \\ -0.85449 & -0.20731 & 2.59561 & -2.13724 \\ 0.70380 & 0.17075 & -2.13724 & 1.75999 \end{bmatrix}$$

The standard deviations of the best estimate values for positions can be found by solving Equation (3.24), the results are

$$\sigma_{E2} = \pm\ 0.00107 \text{ meters}$$
$$\sigma_{N2} = \pm\ 0.00026 \text{ meters}$$
$$\sigma_{E3} = \pm\ 0.00324 \text{ meters}$$
$$\sigma_{N3} = \pm\ 0.00267 \text{ meters}$$

The values of $A(A^T A)^{-1}A^T$ are

$$10^{-3}\begin{bmatrix} 1.472 & -27.197 & 26.960 & -1.235 & -0.518 & -0.843 & -0.856 \\ -27.197 & 503.1 & -498.7 & 22.855 & 0.348 & 0.480 & 0.473 \\ 26.960 & -498.7 & 494.4 & -22.66 & 0.370 & 0.523 & 0.518 \\ -1.235 & 22.855 & -22.66 & 1.038 & -0.201 & -0.159 & -0.134 \\ -0.518 & 0.348 & 0.370 & -0.201 & 10000 & -0.001 & -0.001 \\ -0.843 & 0.480 & 0.523 & -0.159 & -0.001 & 10000 & -0.001 \\ -0.856 & 0.473 & 0.518 & -0.134 & -0.001 & -0.001 & 10000 \end{bmatrix}$$

The standard deviations of adjusted angles and distances are found by solving Equation (3.25), the results are

$$\sigma_{a1} = \pm\ 0.15917 \text{ seconds}$$
$$\sigma_{a2} = \pm\ 2.94270 \text{ seconds}$$
$$\sigma_{a3} = \pm\ 2.91732 \text{ seconds}$$
$$\sigma_{a4} = \pm\ 0.13370 \text{ seconds}$$
$$\sigma_{d1} = \pm\ 0.00002 \text{ meters}$$
$$\sigma_{d2} = \pm\ 0.00002 \text{ meters}$$
$$\sigma_{d3} = \pm\ 0.00002 \text{ meters}$$

The computations in this subsection were performed by the NPS IBM 3033 computer using 16 decimal places and were rounded off to 5 decimal places prior to output.

3. Least Squares Adjustment by the Condition Equation Method

The general principle of least squares adjustment by the condition equation method in surveying is

> to minimize a function consisting of the sum of the squares of the corrections to the observations plus the necessary mathematical conditions involving some or all the corrections; each condition by itself is made equal to zero by adding the corrections to the discrepancy determined from a preadjusted calculation (i.e. calculated from the observed values, rather than from adjusted values); thus, the magnitude of the sum of the squares is not changed by adding conditions [Ref. 8].

For uncorrelated observations of unequal precision, the principle condition of least squares function may be expressed in matrix form as follows:

$$
\begin{aligned}
U &= V^T P V - 2K[BV + W] \\
&= V^T P V - 2[BV + W]^T K \\
&= V^T P V - 2[(BV)^T + W^T]K \\
&= V^T P V - 2[V^T B^T + W^T]K \\
&= V^T P V - 2V^T B^T K + 2W^T K \\
&= \text{minimum}
\end{aligned}
$$

where U is the least squares function matrix, K is a matrix of Lagrange multipliers, P is the weight matrix, V is the correction vector, B is the constant coefficients of the corrections, and W is the precalculated discrepancy [Ref. 9]. It is numerically more convenient for later development to multiply by 2. Taking the partial derivatives of the U matrix with respect to each of the corrections and equating to zero leads to

$$\frac{\partial U}{\partial V} = 2PV - 2B^TK = 0$$

or

$$V = P^{-1}B^TK \tag{3.26}$$

This equation is called a correlate equation or correlate equation matrix. The solution of the Lagrange multipliers matrix can be derived by multiplying Equation (3.26) by the B matrix then adding the W matrix

$$BV + W = BP^{-1}B^TK + W$$

$(BV + W)$ is the condition equation matrix which must equal zero. The solution of the Lagrange multiplier vector is

$$BP^{-1}B^TK = -W$$

$$K = (BP^{-1}B^T)^{-1}(-W) \tag{3.27}$$

The correction vector is derived by substituting Equation (3.27) into Equation (3.26)

$$V = P^{-1}B^T(BP^{-1}B^T)^{-1}(-W) \tag{3.28}$$

For uncorrelated observations of equal precision, the correction vector can be derived similarly to the unequal precision case or by using the inverse of the weight matrix which is the identity matrix. Equation (3.28) becomes

$$V = B^T(BB^T)^{-1}(-W) \tag{3.29}$$

Closed traverse station positions may be adjusted by using the technique of least squares adjustment by the Condition Equation Method. There are two condition equations. They are the azimuth and coordinate condition equations. The coordinate condition equations are divided into two parts, which are E and N coordinate condition equations.

The azimuth condition equation is the sum of the corrections to the angles in a traverse plus a precalculated discrepancy W_1 (Equation 3.6) which must equal zero. Where v_{ai} equals the corrections to the angles, the general equation can be written:

$$\sum_{i=1}^{n} v_{ai} + W_1 = 0$$

or

$$\sum_{i=1}^{n} v_{ai} + \sum_{i=1}^{n-1} 0 \cdot v_{di} + W_1 = 0 \qquad (3.30)$$

where v_{di} is the correction to the distances. However, this term equals zero and does not effect the equation, but it does reserve space in matrix notation for distance corrections. A closed traverse which was conducted at Moss Landing can be written with an azimuth condition of

$$1 \cdot v_{a1} + 1 \cdot v_{a2} + 1 \cdot v_{a3} + 1 \cdot v_{a4} + 0 \cdot v_{d1} + 0 \cdot v_{d2} + 0 \cdot v_{d3} + W_1 = 0 \qquad (3.31)$$

where the precalculated discrepancy (W_1) is +8.844" or +0.00004288 radians.

The simple linearized form of an E coordinate condition equation is the sum of the adjusted departures and must equal zero. By applying Equations (3.2) and (3.7), the general formula for E coordinate condition equation is

$$\sum_{i=1}^{n-1} (d_i + v_{di}) \sin Az_{ai} - (E_T - E_1) = 0 \qquad (3.32)$$

where

$$Az_{ai} = Az_F + \sum_{j=1}^{i} (\alpha_j + v_{aj}) - (i - 1)180^\circ \qquad (3.33)$$

Az_{ai} is the adjusted azimuth. By using Equations (3.32) and (3.33) with the data conducted at Moss Landing, these equations become

$$(d_1 + v_{d1}) \cdot \sin Az_{a1} + (d_2 + v_{d2}) \cdot \sin Az_{a2} + (d_3 + v_{d3}) \cdot \sin Az_{a3} - (E_T - E_1) = 0 \qquad (3.34)$$

where

$$Az_{a1} = Az_F + \alpha_1 + v_{a1}$$

$$= Az_1 + v_{a1}$$

$$Az_{a2} = Az_F + \alpha_1 + v_{a1} + \alpha_2 + v_{a2} - 180^\circ$$

$$= Az_2 + v_{a1} + v_{a2}$$

$$Az_{a3} = Az_F + \alpha_1 + v_{a1} + \alpha_2 + v_{a2} + \alpha_3 + v_{a3} - 360^\circ$$

$$= Az_3 + v_{a1} + v_{a2} + v_{a3}$$

Az_i is the precalculated azimuth or unadjusted azimuth (Equation 3.1).

From trigonometry

$$\sin(A + \triangle A) = \sin A.\cos\triangle A + \cos A.\sin\triangle A$$

in which $\triangle A$ is a very small angle in radians, let

$$\cos\triangle A \doteq 1 \quad \text{and} \quad \sin\triangle A \doteq \triangle A$$

therefore,

$$\sin(A + \triangle A) = \sin A + \triangle A.\cos A$$

from the first term of Equation (3.34)

$$(d_1 + v_{d1})\sin Az_{a1}$$

$$= (d_1 + v_{d1})\sin(Az_1 + v_{a1})$$

$$= (d_1 + v_{d1})[\sin Az_1 + v_{a1}\cos Az_1]$$

$$= d_1\sin Az_1 + v_{a1}d_1\cos Az_1 + v_{d1}\sin Az_1 + v_{d1}v_{a1}\cos Az_1 \tag{3.35}$$

$(v_{d1} \cdot v_{a1})$ is very small, by letting this term equal zero, Equation (3.35) becomes

$$(d_1 + v_{d1})\sin Az_{a1}$$

$$= d_1\sin Az_1 + v_{a1}d_1\cos Az_1 + v_{d1}\sin Az_1 \tag{3.36}$$

The second and third terms of Equation (3.34) can be derived similar to Equation (3.36), so

$$(d_2 + v_{d2})\sin Az_{a2}$$

$$= d_2 \sin Az_2 + (v_{a1} + v_{a2}) d_2 \cos Az_2 + v_{d2} \sin Az_2 \qquad (3.37)$$

$$(d_3 + v_{d3})\sin Az_{a3}$$

$$= d_3 \sin Az_3 + (v_{a1} + v_{a2} + v_{a3}) d_3 \cos Az_3 + v_{d3} \sin Az_3 \qquad (3.38)$$

Substituting Equations (3.36), (3.37), and (3.38) into Equation (3.34) and rearranging terms

$$v_{a1} (d_1 \cos Az_1 + d_2 \cos Az_2 + d_3 \cos Az_3)$$

$$+ v_{a2} (d_2 \cos Az_2 + d_3 \cos Az_3) + v_{a3} d_3 \cos Az_3$$

$$+ v_{d1} \sin Az_1 + v_{d2} \sin Az_2 + v_{d3} \sin Az_3$$

$$+ d_1 \sin Az_1 + d_2 \sin Az_2 + d_3 \sin Az_3 - (E_T - E_1) = 0$$

This equation can be written in the general formula as

$$\sum_{i=1}^{n-1} (v_{ai} \sum_{k=i}^{n-1} d_k \cos Az_k) + \sum_{i=1}^{n-1} v_{di} \sin Az_i + \sum_{i=1}^{n-1} d_i \sin Az_i$$
$$- (E_T - E_1) = 0 \qquad (3.39)$$

substituting Equations (3.2), (3.3), and (3.7) into Equation (3.39)

$$\sum_{i=1}^{n-1} (v_{ai} \sum_{k=i}^{n-1} \triangle N_k) + \sum_{i=1}^{n-1} v_{di} \sin Az_i + W_2 = 0 \qquad (3.40)$$

For example, if the number of the observed angles (n) at Moss Landing is four, this equation can be expanded to

$$v_{a1} (\triangle N_1 + \triangle N_2 + \triangle N_3) + v_{a2} (\triangle N_2 + \triangle N_3) + v_{a3} \triangle N_3$$

$$+ v_{d1} \sin Az_1 + v_{d2} \sin Az_2 + v_{d3} \sin Az_3 + W_2 = 0$$

Substituting the numerical values from the precalculated of this traverse into this equation, the result is

$$(6702.37612)v_{a1} + (5318.48345)v_{a2} + (4999.28284)v_{a3} + 0.v_{a4}$$

$$-(0.23567)v_{d1} + (0.48818)v_{d2} + (0.63570)v_{d3} + W_2 = 0 \qquad (3.41)$$

where the precalculated discrepancy (W_2) is 0.08608 m and $0.v_{a4}$ is equal to zero, which does not effect this equation, but space is reserved in matrix notation for the last angle correction.

The N coordinate condition equation is the sum of the adjusted latitudes and must equal zero. By applying Equations (3.3) and (3.7), the general condition equation is

$$\sum_{i=1}^{n-1}(d_i + v_{d_i}) \cos Az_{a_i} - (N_T - N_1) = 0$$

This equation can be derived similarly to the E coordinate condition equation case, resulting in

$$-\sum_{i=1}^{n-1}(v_{a_i} \sum_{k=1}^{n-1}\triangle E_k) + \sum_{i=1}^{n-1} v_{d_i} \cos Az_i + W_3 = 0 \qquad (3.42)$$

From the data acquired at Moss Landing, this equation can be expanded to

$$-v_{a1}(\triangle E_1 + \triangle E_2 + \triangle E_3) - v_{a2}(\triangle E_2 + \triangle E_3) + v_{a3}\triangle E_3$$

$$+ v_{d1}\cos Az_1 + v_{d2}\cos Az_2 + v_{d3}\cos Az_3 + W_3 = 0$$

Substituting the numerical values from the calculation of this traverse into this equation, the result is

$$-(3959.89460)v_{a1} - (4295.49626)v_{a2} - (4116.94755)v_{a3} + 0.v_{a4}$$

$$+(0.97183)v_{d1} + (0.87274)v_{d2} + (0.77194)v_{d3} + W_3 = 0 \qquad (3.43)$$

where the precalculated discrepancy (W_3) is -0.08936 m.

Equations (3.30), (3.40), and (3.42) can be written in the matrix form as (BV + W), where B is the constant coefficients of the corrections, V is the correction vector which is composed of the angle and distance corrections, W is the precalculated discrepancy. Equations (3.31), (3.41), and (3.43) are written in matrix form as

$$B^T = \begin{bmatrix} 1 & 6702.37612 & -3959.89460 \\ 1 & 5318.48345 & -4295.49626 \\ 1 & 4999.28284 & -4116.94755 \\ 1 & 0.00000 & 0.00000 \\ 0 & -0.23567 & 0.97183 \\ 0 & 0.48818 & 0.87274 \\ 0 & 0.63570 & 0.77194 \end{bmatrix} \qquad V = \begin{bmatrix} v_{a1} \\ v_{a2} \\ v_{a3} \\ v_{a4} \\ v_{d1} \\ v_{d2} \\ v_{d3} \end{bmatrix}$$

$$-W = \begin{bmatrix} -W_1 = -0.00004 \\ -W_2 = -0.08608 \\ -W_3 = \ \ 0.08936 \end{bmatrix}$$

For uncorrelated observations of unequal precision, the inverse of the weight matrix is the diagonal matrix, which is equal to the observation's variance. The diagonal elements of the inverse of the weight matrix is

$$P_{11}^{-1} = \sin^2(1.984'') = 0.92519 \times 10^{-10}$$

$$P_{22}^{-1} = \sin^2(1.405'') = 0.46398 \times 10^{-10}$$

$$P_{33}^{-1} = \sin^2(1.203'') = 0.34016 \times 10^{-10}$$

$$P_{44}^{-1} = \sin^2(1.614'') = 0.61229 \times 10^{-10}$$

$$P_{55}^{-1} = (0.001)^2 = 0.10000 \times 10^{-5}$$

$$P_{66}^{-1} = (0.001)^2 = 0.10000 \times 10^{-5}$$

$$P_{77}^{-1} = (0.003)^2 = 0.90000 \times 10^{-5}$$

For uncorrelated observations of unequal precision, the correction vector is found by solving Equation (3.28)

$$V = \begin{bmatrix} v_{a1} = 0.93141 \times 10^{-5} \text{ radians or } 1.92117'' \\ v_{a2} = -1.63114 \times 10^{-5} \text{ radians or } -3.36446'' \\ v_{a3} = -1.28358 \times 10^{-5} \text{ radians or } -2.64758'' \\ v_{a4} = -2.30438 \times 10^{-5} \text{ radians or } -4.75312'' \\ v_{d1} = 0.00024 \text{ meters} \\ v_{d2} = 0.00039 \text{ meters} \\ v_{d3} = 0.00357 \text{ meters} \end{bmatrix}$$

The corrected angles are

$$\alpha_1 = 246^\circ\ 05'\ 45.12117''$$
$$\alpha_2 = 222^\circ\ 51'\ 05.23554''$$
$$\alpha_3 = 190^\circ\ 14'\ 59.95242''$$
$$\alpha_4 = 277^\circ\ 05'\ 12.24688''$$

The corrected distances are

$$d_1 = 1424.00424 \text{ m.}$$
$$d_2 = 365.74439 \text{ m.}$$
$$d_3 = 6476.27457 \text{ m.}$$

Using the corrected data to recompute the coordinates is similar to the initial traverse computations. The adjusted coordinates are

$$E_2 = 607{,}943.45521 \text{ m.}$$
$$N_2 = 4{,}073{,}939.74809 \text{ m.}$$
$$E_3 = 608{,}122.00189 \text{ m.}$$
$$N_3 = 4{,}074{,}258.95029 \text{ m.}$$
$$E_4 = 612{,}238.85256 \text{ m.}$$
$$N_4 = 4{,}079{,}258.31754 \text{ m.}$$

The last adjusted coordinates are the same values as the fixed coordinates. Hence, it is not necessary to iterate. For uncorrelated observations with equal precision, the correction vector is found by solving Equation (3.29)

$$
V = \begin{bmatrix}
v_{a1} & = & 1.26884 \times 10^{-5} & \text{radians or} & 2.61717'' \\
v_{a2} & = & -1.56886 \times 10^{-5} & \text{radians or} & -3.23601'' \\
v_{a3} & = & -1.75408 \times 10^{-5} & \text{radians or} & -3.61804'' \\
v_{a4} & = & -2.23360 \times 10^{-5} & \text{radians or} & -4.60713'' \\
v_{d1} & = & 0.14034 \times 10^{-7} & \text{meters} & \\
v_{d2} & = & 0.23842 \times 10^{-7} & \text{meters} & \\
v_{d3} & = & 0.24362 \times 10^{-7} & \text{meters} &
\end{bmatrix}
$$

The corrected angles are

$$\alpha_1 = 246^\circ\ 05'\ 45.81717''$$
$$\alpha_2 = 222^\circ\ 51'\ 05.36399''$$
$$\alpha_3 = 190^\circ\ 14'\ 58.98196''$$
$$\alpha_4 = 277^\circ\ 05'\ 12.39287''$$

The corrected distances are

$$d_1 = 1424.00400 \text{ m.}$$
$$d_2 = 365.74400 \text{ m.}$$
$$d_3 = 6476.27100 \text{ m.}$$

By using the corrected data to recompute the coordinates, the adjusted coordinates are

$$E_2 = 607{,}943.45994 \text{ m.}$$
$$N_2 = 4{,}073{,}939.74899 \text{ m.}$$
$$E_3 = 608{,}122.00770 \text{ m.}$$
$$N_3 = 4{,}074{,}258.95013 \text{ m.}$$
$$E_4 = 612{,}238.85256 \text{ m.}$$
$$N_4 = 4{,}079{,}258.31754 \text{ m.}$$

The last adjusted coordinates are the same values as the fixed coordinates. Hence, it is not necessary to iterate.

The standard deviation of an observation which has unit weight can be found by the following equations. For uncorrelated observations of unequal precision,

$$\sigma_o = \left[\frac{V^T P V}{r}\right]^{1/2} \tag{3.44}$$

and, for uncorrelated observations of equal precision,

$$\sigma_o = \left[\frac{V^T V}{r}\right]^{1/2} \tag{3.45}$$

where r is the number of condition equations. There are three condition equations for a closed traverse (r=3). The standard deviations of adjusted angles and distances can be found by the following equation:

$$\sigma_i = \sigma_o \left[Q_{ii} \right]^{1/2} \tag{3.46}$$

where σ_i is the standard deviation of the ith adjusted quantity. The quantity in the ith row of the V matrix, Q_{ii} is an element of the $P^{-1} - P^{-1}B^T(BP^{-1}B^T)^{-1}BP^{-1}$ matrix for uncorrelated observations of unequal precision [Ref. 3, p. 918]. For uncorrelated observations of equal precision, Q_{ii} is the element of the $I - B^T(BB^T)^{-1}B$ matrix. The matrix $P^{-1} - P^{-1}B^T(BP^{-1}B^T)^{-1}BP^{-1}$ or the matrix $I - B^T(BB^T)^{-1}B$ is equal to

$$\begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1n} \\ Q_{21} & Q_{22} & \cdots & Q_{2n} \\ \vdots & \vdots & & \vdots \\ Q_{n1} & Q_{n2} & \cdots & Q_{nn} \end{bmatrix}.$$

For uncorrelated observations of unequal precision, the standard deviation of unit weight is found by solving Equation (3.44), the result is

$$\sigma_\circ = \pm\ 2.69679$$

The results of $P^{-1} - P^{-1}B^T(BP^{-1}B^T)^{-1}BP^{-1}$ are

$$10^{-10}\begin{bmatrix} 0.064 & -0.052 & -0.021 & 0.010 & -4.165 & -6.963 & -63.88 \\ -0.052 & 0.225 & -0.175 & 0.002 & 3.415 & 4.689 & 41.59 \\ -0.021 & -0.175 & 0.211 & -0.014 & 2.563 & 3.574 & 31.79 \\ 0.010 & 0.002 & -0.014 & 0.002 & -1.813 & -1.301 & -9.500 \\ -4.165 & 3.415 & 2.563 & -1.813 & 9917 & -117.4 & -1047 \\ -6.963 & 4.689 & 3.574 & -1.301 & -117.4 & 9830 & -1525 \\ -63.88 & 41.59 & 31.79 & -9.500 & -1047 & -1525 & 76323 \end{bmatrix}$$

The standard deviations of adjusted angles and distances are found by solving Equation (3.46); the results are

$$\begin{aligned} \sigma_{a1} &= \pm\ 1.40340 \text{ seconds} \\ \sigma_{a2} &= \pm\ 2.64096 \text{ seconds} \\ \sigma_{a3} &= \pm\ 2.55253 \text{ seconds} \\ \sigma_{a4} &= \pm\ 0.26136 \text{ seconds} \\ \sigma_{d1} &= \pm\ 0.00269 \text{ meters} \\ \sigma_{d2} &= \pm\ 0.00267 \text{ meters} \\ \sigma_{d3} &= \pm\ 0.00745 \text{ meters} \end{aligned}$$

For uncorrelated observations of equal precision, the standard deviation of unit weight is found by solving Equation (3.45), the result is

$$\sigma_\circ = \pm\ 2.01139 \times 10^{-5}$$

The results of $I - B^T(BB^T)^{-1}B$ are

$$10^{-3}\begin{bmatrix} 1.472 & -27.199 & 26.962 & -1.235 & -0.518 & -0.843 & -0.856 \\ -27.199 & 503.1 & -498.7 & 22.855 & 0.348 & 0.480 & 0.473 \\ 26.962 & -498.7 & 494.4 & -22.66 & 0.370 & 0.523 & 0.518 \\ -1.235 & 22.855 & -22.66 & 1.038 & -0.201 & -0.159 & -0.134 \\ -0.518 & 0.348 & 0.370 & -0.201 & 10000 & -0.001 & -0.001 \\ -0.843 & 0.480 & 0.523 & -0.159 & -0.001 & 10000 & -0.001 \\ -0.856 & 0.473 & 0.518 & -0.134 & -0.001 & -0.001 & 10000 \end{bmatrix}$$

The standard deviations of adjusted angles and distances are found by solving Equation (3.46); the results are

$$\begin{aligned} \sigma_{a1} &= \pm\ 0.15918 \text{ seconds} \\ \sigma_{a2} &= \pm\ 2.94262 \text{ seconds} \\ \sigma_{a3} &= \pm\ 2.91722 \text{ seconds} \\ \sigma_{a4} &= \pm\ 0.13369 \text{ seconds} \\ \sigma_{d1} &= \pm\ 0.00002 \text{ meters} \\ \sigma_{d2} &= \pm\ 0.00002 \text{ meters} \\ \sigma_{d3} &= \pm\ 0.00002 \text{ meters} \end{aligned}$$

The computations in this subsection were also performed on the NPS IBM 3033 computer; they used 16 decimal places, and were rounded off to 5 decimal places in the final solutions. These are more decimal places than are normally used in practice because such precisions are not generally attainable by corresponding observations. However, in comparing two computational methods that are theoretically equal, one method may be more sensitive to round off error than the other. This will be commented on in the next chapter.

IV. DISCUSSIONS AND ANALYSIS OF RESULTS

Three programs were used to compute the initial traverse and adjust the closed traverse station positions. Programs 1, 2, and 3 were used to compute and adjust the traverse station positions by the Approximate Method, the Indirect Observations Method, and the Condition Equation Method, respectively. The programs were written in WATFIV language for implementation on the NPS IBM 3033 computer. The computer output and listings of Programs 1, 2, and 3 are provided in Appendices A, B, and C, respectively. The maximum number of intermediate traverse station positions that can be computed and adjusted by these programs is 30. The programs were tested using several fictitious data sets to ensure their performance in handling the various intermediate traverse station positions.

The computer storage area and CPU time of Programs 1, 2, and 3 has been compared (Table X). The adjustment of a closed traverse by the Approximate Method did not use a weight matrix in the matrix computations. The computer program for this method was written by using the variables in only one dimension for storage of both data and results. This method did not require any iterations. The least squares adjustment of a closed traverse characteristically is used to simultaneously eliminate closing errors in azimuths and distances, and Programs 2 and 3 both utilized such an adjustment. The computation by either Least Squares Method used matrices in two dimensions to compute the correction vector. Likewise the computer programs were written using variables in two dimensions. Included are all subroutines from Program 1 and extra subroutines for each individual program. Therefore, both Programs 2 and 3 used more computer storage area and CPU time than Program 1.

TABLE X

The Comparisons Between the Computer Storage Area and CPU Time of Programs 1, 2, and 3

	Programs			Unit
	1	2	3	
1. The number of statements	450	874	648	
2. The ratio of statements of Programs 1:2 and 1:3	1:1	1:1.9	1:1.4	
3. The ratio of statements of Programs 3:2	-	1:1.3	1:1	
4. The total computer storage area	25,536	294,856	129,728	Bytes
5. The ratio of computer storage area of Programs 1:2 and 1:3	1:1	1:11.5	1:5.1	
6. The ratio of computer storage area of Programs 3:2	-	1:2.3	1:1	
7. The CPU time for adjusted two stations	.62	1.56	1.10	Sec.
8. The ratio of CPU time of Programs 1:2 and 1:3	1:1	1:2.5	1:1.8	
9. The ratio of CPU time of Programs 3:2	-	1:1.4	1:1	

The Indirect Observations Method requires that the number of observation equations be equal to the sum of observation equations of both observed angles and distances. The number of observation equations is not constant. It changes depending on the number of the traverse stations--this means that the row dimension of each matrix will change depending on the number of observation equations. Therefore, the subroutines affected by the number of traverse stations are difficult to write in the general program. The coefficients in the A matrix are computed by taking the partial derivatives of the functions with respect to each unknown variable, where the number of unknown variables is not constant. The number of unknown variables changes depending on the number of the traverse stations. Therefore, the subroutine for the computed A matrix is also difficult to write in the general program. The Condition Equations Method has only three equations which makes it easier to derive the general form and write the computer program. The adjustment using the Indirect Observations Method does not apply the corrections to the observed values directly; therefore, it requires the approximate values of the unknown coordinates be computed for the correction vector of the unknown coordinates. This method requires at least two iterations to check the insignificance of the correction vector when it is compared with an arbitrarily selected small number. The adjusted coordinates from the Condition Equations Method can be checked at the first iteration. Thus, Program 3 is more economical as it uses less computer storage area and CPU time than Program 2 (Table X).

The closed traverse at Moss Landing originated and terminated at control points with known positions which were determined by third-order methods. The azimuth closure was 2.211" per station and the position closure was 1:66,617. The classification, standards of accuracy, and general

specifications for a third-order class I traverse (Table I) indicate the azimuth closure is not to exceed 3" per station and position closure must be better than 1:10,000. This traverse met the specifications and standards of accuracy for a third-order class I traverse.

Using Programs 1, 2, and 3, the traverse was computed and adjusted in UTM grid coordinates. The final adjusted coordinates were transformed from UTM grid coordinates to geographic coordinates [Ref. 4, pp. 319-321]. The difference between the coordinates for each method and NOS [Ref. 10] have been computed (Table XI). The technique of Least Squares for both methods yields identical computational results to at least five decimal places (Table XI, Methods 3.1 and 4.1; 3.2 and 4.2). Least Squares provides the best estimates for positions of all traverse stations. Program 2 performs a statistical test, yielding the standard deviations of both adjusted positions and observed values. Conversely, Program 3 only yields the standard deviation of adjusted observed values. Comparisons of the standard deviations of adjusted observed angles and distances were made between these computed by the Indirect Observations Method and the Condition Equations Method (Table XII). The computed standard deviations of the adjusted angles differed significantly in the fourth decimal place in several cases, which is a slight indication that the Indirect Observations Method is more sensitive to round off error. The standard deviation estimates are larger for the Indirect Observations Method in every case of a significant difference in the fourth decimal place. However, a significant difference in the fourth decimal place is insignificant in terms of the observational precision. The standard deviations of the adjusted distances for both methods yield identical results (Table XII). Both Methods differ from the NOS positions in the fourth decimal place in seconds of arc for both latitude

and longitude. The horizontal control for third-order standards requires accuracy to three decimal places in seconds of arc for both latitude and longitude--thus, these methods can be used for computing third-order positions.

TABLE XI

Geographic Coordinates

	At Mossback station			
Methods	Latitude (N.)	Diff.	Longitude (W.)	Diff.
	° ' "	"	° ' "	"
1. U.S. NOS	36 48 25.09750	-	121 47 23.75903	-
2. Approximate	36 48 25.09713	+0.00037	121 47 23.75934	-0.00031
3. Indirect Observations				
3.1 Equal precision	36 48 25.09762	-0.00012	121 47 23.75869	+0.00034
3.2 Unequal precision	36 48 25.09759	-0.00009	121 47 23.75889	+0.00014
4. Condition Equation				
4.1 Equal precision	36 48 25.09762	-0.00012	121 47 23.75869	+0.00034
4.2 Unequal precision	36 48 25.09759	-0.00009	121 47 23.75889	+0.00014
	At Dune Temp station			
Methods	Latitude (N.)	Diff.	Longitude (W.)	Diff.
	° ' "	"	° ' "	"
1. U.S. NOS	36 48 35.38113	-	121 47 16.39167	-
2. Approximate	36 48 35.38077	+0.00036	121 47 16.39200	-0.00033
3. Indirect Observations				
3.1 Equal precision	36 48 35.38121	-0.00008	121 47 16.39129	+0.00038
3.2 Unequal precision	36 48 35.38122	-0.00009	121 47 16.39152	+0.00015
4. Condition Equation				
4.1 Equal precision	36 48 35.38121	-0.00008	121 47 16.39129	+0.00038
4.2 Unequal precision	36 48 35.38122	-0.00009	121 47 16.39152	+0.00015

TABLE XII

The Comparisons Between the Standard Deviations of Least Squares

	Unequal Precision Observations						
	Standard Deviations of						
	Adjusted Angles				Adjusted Distances		
	1	2	3	4	1	2	3
Methods	"	"	"	"	m.	m.	m.
1. Condition Equation	1.40340	2.64096	2.55253	0.26136	0.00269	0.00267	0.00745
2. Indirect Observation	1.40346	2.64103	2.55260	0.26137	0.00269	0.00267	0.00745
Difference	-.00006	-.00007	-.00007	-.00001	-	-	-

	Equal Precision Observations						
	Standard Deviations of						
	Adjusted Angles				Adjusted Distances		
	1	2	3	4	1	2	3
Methods	"	"	"	"	m.	m.	m.
1. Condition Equation	0.15918	2.94262	2.91722	0.13369	0.00002	0.00002	0.00002
2. Indirect Observation	0.15917	2.94270	2.91732	0.13370	0.00002	0.00002	0.00002
Difference	+.00001	-.00008	-.00010	-.00001	-	-	-

V. CONCLUSIONS AND RECOMMENDATION

A. CONCLUSIONS

The computer programs developed by the author and contained in this thesis have a variety of useful and practical hydrographic applications. In hydrography, geodetic field work plays a key role. Horizontal control is necessary in determining positions and, even in areas which appear to initially have adequate control, additional control stations often need to be established after arrival in the field. These field-generated control points must be adjusted to fit within the existing survey net and to minimize errors in the field measurements. The errors can be minimized by these computer programs. Direct application of these programs would be enormously beneficial, since no equivalent software exist at NPS to perform such adjustments at the present time.

In the Approximate Method, a traverse is adjusted by computations using a hand calculator. This method is suitable for field computation and the adjusted coordinates meet the specifications and standards of accuracy for a third-order class I traverse. Although the accuracy from this method is less than the Least Squares Metnod, it does not require a computer and it furnishes coordinates which can be checked in the field by the field party.

The Least Squares Method provides the best estimates for positions of all traverse stations. Least Squares does require more of a computational effort than the Approximate Method. However, the accuracy of Least Squares is better than the Approximate Method, and the Least Squares Method should be used for the final office computations. Both

Least Squares Methods yield identical results, but the Condition Equations Method is more economical and easier to derive than the Indirect Observations Method. Therefore, the Condition Equations Method should be applied at NPS.

B. RECOMMENDATION

Studies should be continued at NPS to compare the economy of adjustment methods used in this thesis with other methods of least squares adjustment.

APPENDIX A

PROGRAM 1

COMPUTER OUTPUT

THE CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
BY APPROXIMATE METHOD

KNOWN HORIZONTAL STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM NO	TO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	PIPHER					
2	MOSS 2	4072555.85206	608279.04404	2 ->	1	100 16 23.77800
3	HOLM	4079258.31754	612238.85256	3 ->	4	136 33 26.33400
4	MORAN					

OBSERVED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	STD. S.	GRID DISTANCE METERS	STD. M.
PIPHER				
MOSS 2	246 5 43.20000	1.98400		
			1424.00400	0.00100
MOSSBACK	222 51 8.60000	1.40500		
			365.74400	0.00100
DUNE TEMP	190 15 2.60000	1.20300		
			6476.27100	0.00300
HOLM	277 5 17.00000	1.61400		
MORAN				

PRECALCULATED OF CLOSED TRAVERSE STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	UTM GRID COORDINATES GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 6.97800
2	MOSSBACK	4073939.74473	607943.44238	2 -> 1	166 22 6.97800
				2 -> 3	29 13 15.57800
3	DUNE TEMP	4074258.94534	608121.99110	3 -> 2	209 13 15.57800
				3 -> 4	39 28 18.17800
4	HOLM	4079258.22818	612238.93865	4 -> 3	219 28 18.17800
				4 -> 5	136 33 35.17800
5	MORAN				

TOTAL AZIMUTH ERROP = 0 D. 0 M. 8.84400 S.
AZIMUTH ERROR PER STATIONS = 0 D. 0 M. 2.21100 S.
TOTAL GRID DISTANCE = 8266.01900 METERS.
DISTANCE ERROR = 0.12408 METERS.
ACCURACY = 1: 66617

ADJUSTMENT OF CLOSED TRAVERSE STATION POSITIONS BY APPROXIMATE METHOD

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 4.76700
2	MOSSBACK	4073939.73368	607943.44415	2 -> 1	166 22 4.76700
				2 -> 3	29 13 11.15600
3	DUNE TEMP	4074258.93620	608121.99028	3 -> 2	209 13 11.15600
				3 -> 4	39 28 11.54500
4	HOLM	4079258.31754	612238.85256	4 -> 3	219 28 11.54500
				4 -> 5	136 33 26.33400
5	MORAN				

CHECK ROUND OFF ERROR

AZIMUTH = 0 D. 0 M. 0.00000 S.

DISTANCE = 0.00000 METERS.

PROGRAM LISTING

```
C               TRAVAPP WATFIV A1
C
$JOB            SAM1,NOCHECK
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN.                              C
C        DATE : JULY 11 , 1984                                        C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     P
C        THIS IS A PROGRAM FOR CALCULATION  OF TRAVERSE STATION       P
C   POSITIONS BY APPROXIMATE METHOD.                                  P
C                                                                     P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                     V
C                VARIABLE DEFINITIONS                                 V
C                                                                     V
C     OPT      = OPTION FOR CALCULATION OF OPEN OR CLOSED TRAVERSE    V
C     CHOIC    = OPTION FOR ENTER DATA BY DATA FILE OR INTERACTIVE    V
C                FROM TERMINAL KEYBOARD.                              V
C     DUMMY1,DUMMY2,DUM1D,DUM1M,DUM1S = THE DUMMY VARIABLES           V
C     NGRID(4)= GRID NORTHING OF THE KNOWN POSITIONS.                 V
C     EGRID(4)= GRID EASTING OF THE KNOWN POSITIONS.                  V
C     STNAME(4,5) = THE NAME OF KNOWN STATIONS.                       V
C     TRNAME( 32,5) = THE NAME OF TRAVERSE STATIONS.                  V
C     DIRECD(4),DIRECM(4),DIRECS(4) = THE GRID AZIMUTH OF KNOWN       V
C                POSITIONS.                                           V
C     STAT1(4),STAT2(4) = THE NUMBER OF KNOWN POSITIONS.              V
C     COUNT, COUNTA = THE NUMBER OF OBSERVED ANGLES.                  V
C     ANG( 32)= THE OBSERVED ANGLE IN RADIAN.                         V
C     ANGD( 32),ANGM( 32),ANGS( 32) = THE OBSERVED ANGLES IN DEGREE,  V
C                MINUTE AND SECOND.                                   V
C     COUNTN, COUNTD = THE NUMBER OF MEASURED DISTANCE.               V
C     DIST( 32) = THE MEASURED DISTANCE IN METERS.                    V
C     TODIS    = THE SUM OF MEASURED DISTANCE.                        V
C     CFAZD( 32),CFAZM( 32),CFAZS( 32) = THE COMPUTED FORWARD AZIMUTH V
C     CBAZD( 32),CBAZM( 32),CBAZS( 32) = THE COMPUTED BACK AZIMUTH    V
C     CNGRID( 32) = GRID NORTHING OF TRAVERSE STATION POSITIONS.      V
C     CEGRID( 32) = GRID EASTING OF TRAVERSE STATION POSITIONS.       V
C     ERR      = THE TOTAL DISTANCE ERROR.                            V
C     DIFAZI   = THE TOTAL ANGLE ERROR IN RADIAN.                     V
C     EAZ1D, EAZ1M, EAZ1S = THE TOTAL ANGLE ERROR IN DEGREE,          V
C                MINUTE AND SECOND.                                   V
C     EAZ2D, EAZ2M, EAZ2S = THE ANGLE ERROR PER STATION.              V
```

```
C       TDMS      = SUBROUTINE FOR CHANGE INPUT DATA FORM DDD.MMSSSSSSS     V
C                   TO DEGREE , MINUTE AND SECOND.                          V
C       CDMSR     = SUBROUTINE FOR CHANGE DEGREE, MINUTE AND                V
C                   SECOND TO RADIAN.                                       V
C       CRDMS     = SUBROUTINE FOR CHANGE RADIAN TO DEGREE, MINUTE          V
C                   AND SECOND.                                             V
C       CTANA     = SUBROUTINE FOR CALCULATION OF GRID AZIMUTH AND THE      V
C                   THE DISTANCE BETWEEN POSITIONS # 1 AND #2               V
C       CXYC      = SUBROUTINE FOR COMPUTATION OF THE UTM GRID COORDINATES  V
C                                                                           V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                           C
C                       VARIABLE DECLARATIONS                               C
C                                                                           C
      DOUBLE PRECISION DUMMY1,CFAZS( 32),CBAZS( 32),EAZ1S,EAZ2S,CEGRID(
     *32),CNGRID( 32),DIS21,AZ21,AZ12,DIS34,AZ34,DUM1D,AZ43,DUM1M,DUM1S,
     *DUMMY2,ANG( 32),DIST( 32),AZFIR,AZIMUT,NUMO,NUM360,AN360R,DISTAN,
     *AZC,DIFDIS,AZLAS,DIFAZI,CORAZ,TODIS,COAZR( 32),WPRED(3),SUMDY
      DOUBLE PRECISION ANGS( 32),DIRECS(4),NGRID(4),EGRID(4),CORDX1,CORD
     *Y1,XGD,YGD,STDA( 32),STDD( 32),DELTAX( 32),DELTAY( 32),SUMDX
C
      INTEGER DES,OPT,COUNT,I,COUNTA,COUNTD,ERR,CHOIC,LOOP1,DIRECD(4),DI
     *RECM(4),STAT1(4),STAT2(4),STNAME(4,5),K1,K2,ANGD( 32),ANGM( 32),CO
     *UNTN,TRNAME( 32,5),K4,K5,COUNTP,EAZ1D,EAZ1M,EAZ2D,EAZ2M,CFAZD( 32)
     *,CBAZD( 32),CBAZM( 32),TEMD,TEMM,ADJ1,CFAZM( 32),ITER1
C
      DATA NUM0/0.0D0/,NUM360/360.0D0/,TODIS/0.0D0/,SUMDX/0.0D0/,SUMDY/0
     *.0D0/
C
C
C       ALGORITHM INPUT_DATA
C
C       PRINT TITLE
C
C
      WRITE ( 6,1000 )
      WRITE ( 6,1380 )
      WRITE ( 6,1390 )
      WRITE ( 6,1400 )
      WRITE ( 6,1410 )
      WRITE ( 6,1420 )
      WRITE ( 6,1390 )
      WRITE ( 6,1380 )
      WRITE ( 6,1430 )
      WRITE ( 6,1440 )
C
C
C       READ OPTIONS 1 = OPEN TRAVERSE
```

```
C                    2 = CLOSED TRAVERSE
C
      READ ( 4,1340 ) OPT
C
C
      WRITE ( 6,1450 )
      WRITE ( 6,1460 )
      WRITE ( 6,1470 )
      WRITE ( 6,1430 )
      WRITE ( 6,1480 )
C
C
C      READ OPTIONS 1 = ENTER DATA BY DATA FILE
C                   2 = ENTER DATA BY INTERACTIVE
C
      READ ( 4,1340 ) CHOIC
C
C
C      GET UTM GRID NORTHING, EASTING, AZIMUTH, AND THE NAME
C                      OF KNOWN POSITIONS
C
C
      DES = 04
      IF ( CHOIC .EQ. 2 ) GO TO 10
           DES = 03
 10   CONTINUE
      LOOP1 = 2
      IF ( OPT .EQ. 1 ) GO TO 20
           LOOP1 = 4
 20   CONTINUE
      DO  70  I = 1,LOOP1
          WRITE ( 6,1490 ) I
          IF ( I .NE. 1 ) GO TO 30
               WRITE ( 6,1500 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
               GO TO 60
 30       CONTINUE
          IF ( I .NE. 2 ) GO TO 40
               WRITE ( 6,1520 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
               WRITE ( 6,1660 )
               WRITE ( 6,1540 )
               READ ( DES,1360 ) DUMMY1
               NGRID ( I ) = DUMMY1
               WRITE ( 6,1670 )
               WRITE ( 6,1560 )
               READ ( DES,1360 ) DUMMY1
               EGRID ( I ) = DJMMY1
```

```
              WRITE ( 6,1680 ) (STNAME(2,K1),K1=1,5),(STNAME(1,K1),K1=1
     *        ,5)
              WRITE ( 6,1690 )
              READ ( DES,1360 ) DUMMY1
              CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
              DIRECD ( 2 ) = IDINT ( DUM1D )
              DIRECM ( 2 ) = IDINT ( DUM1M )
              DIRECS ( 2 ) = DUM1S
              CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
              AZ21 = DUMMY1
              STAT1 ( 2 ) = 2
              STAT2 ( 2 ) = 1
              GO TO 60
   40      CONTINUE
           IF ( I .NE. 3 ) GO TO 50
              WRITE ( 6,1530 )
              READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
              WRITE ( 6,1660 )
              WRITE ( 6,1540 )
              READ ( DES,1360 ) DUMMY1
              NGRID ( I ) = DUMMY1
              WRITE ( 6,1670 )
              WRITE ( 6,1560 )
              READ ( DES,1360 ) DUMMY1
              EGRID ( I ) = DUMMY1
              GO TO 60
   50      CONTINUE
           IF ( I .NE. 4 ) GO TO 60
              WRITE ( 6,1510 )
              READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
              WRITE ( 6,1680 ) (STNAME(3,K1),K1=1,5),(STNAME(4,K1),K1=1
     *        ,5)
              WRITE ( 6,1690 )
              READ ( DES,1360 ) DUMMY1
              CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
              DIRECD ( 3 ) = IDINT ( DUM1D )
              DIRECM ( 3 ) = IDINT ( DUM1M )
              DIRECS ( 3 ) = DUM1S
              CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
              AZ34 = DUMMY1
              STAT1 ( 3 ) = 3
              STAT2 ( 3 ) = 4
   60      CONTINUE
   70   CONTINUE
C
C
C       READ THE NUMBER OF OBSERVED ANGLES
C
```

```
C
      WRITE ( 6,1580 )
      READ ( DES,1350 ) COUNT
      COUNTA = COUNT
C
C     READ THE OBSERVED ANGLES
C
      DO  100 I = 1,COUNTA
          IF ( I .GT. 1 ) GO TO 80
               WRITE ( 6,1590 )
               GC TO 90
 80       CONTINUE
          K2 = I -1
          WRITE ( 6,1600 ) K2
 90       CONTINUE
          WRITE ( 6,1610 )
          READ ( DES,1360 ) DUMMY1
          CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
          CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY2 )
          ANG ( I ) = DUMMY2
          ANGD ( I ) = IDINT ( DUM1D )
          ANGM ( I ) = IDINT ( DUM1M )
          ANGS ( I ) = DUM1S
          WRITE ( 6,1550 )
          READ ( DES,1360 ) DUMMY1
          STDA ( I ) = DUMMY1
 100  CONTINUE
C
C
C     READ THE GRID DISTANCE
C
C
      COUNTD = CCUNT
      IF ( OPT .EQ. 1 ) GO TO 110
           COUNTD = COUNT - 1
 110  CONTINUE
      DO  140 I = 1,COUNTD
          IF ( I .GT. 1 ) GO TJ 120
               WRITE ( 6,1620 )
               GO TO 130
 120      CONTINUE
          K4 = I - 1
          K5 = I
          WRITE ( 6,1630 ) K4,K5
 130      CONTINUE
          WRITE ( 6,1640 )
          READ ( CES,1360 ) DUMMY1
          DIST ( I ) = DUMMY1
```

```
          TODIS = TODIS + DUMMY1
          WRITE ( 6,1570 )
          READ ( DES,1360 ) DUMMY1
          STDD ( I ) = DUMMY1
  140  CONTINUE
       COUNTN = COUNTD
       IF ( OPT .EQ. 1 ) GO TO 150
            COUNTN = COUNTD - 1
  150  CONTINUE
       DO  160 I = 1,COUNTN
          WRITE ( 6,1650 ) I
          READ ( DES,1370 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
  160  CONTINUE
C
C
C      END INPUT_DATA
C
C      ALGORITHM PRINT_INPUT_DATA
C
C      PRINT DETAILS OF THE KNOWN STATIONS
C
C
       WRITE ( 6,1000 )
       WRITE ( 6,1010 )
       WRITE ( 6,1020 )
       WRITE ( 6,1030 )
       WRITE ( 6,1040 )
       WRITE ( 6,1050 )
       WRITE ( 6,1060 )
       DO  180 I = 1,LOOP1
          IF ( I .EQ. 2 .OR. I .EQ. 3 ) GO TO 170
               WRITE ( 6,1230 )  I , ( STNAME ( I,K1 ) , K1 = 1,5 )
               GO TO 180
  170     CONTINUE
          WRITE ( 6,1070 ) I,(STNAME(I,K1),K1=1,5),NGRID(I),EGRID(I),STA
     *    T1(I),STAT2(I),DIRECD(I),DIRECM(I),DIRECS(I)
  180  CONTINUE
       WRITE ( 6,1000 )
       WRITE ( 6,1080 )
       WRITE ( 6,1090 )
       WRITE ( 6,1100 )
       WRITE ( 6,1110 )
       WRITE ( 6,1120 )
       WRITE ( 6,1130 ) ( STNAME ( 1,K1 ) , K1 = 1,5 )
       WRITE ( 6,1140 ) (STNAME(2,K1),K1 = 1,5),ANGD(1),ANGM(1),ANGS(1),S
      *TDA(1)
       WRITE ( 6,1150 ) DIST ( 1 ),STDD ( 1 )
       COUNTP = COUNTD - 1
```

```
      IF ( COUNTP .NE. 0 ) GO TO 190
           WRITE ( 6,1160 ) ( TRNAME ( 1,K1 ) , K1 = 1,5 )
           GO TO 220
 190  CONTINUE
      DO  200 I = 1,COUNTP
          K2 = I + 1
          WRITE ( 6,1140 ) (TRNAME(I,K1),K1=1,5),ANGD(K2),ANGM(K2),ANGS(
     *    K2),STDA(K2)
          WRITE ( 6,1150 ) DIST ( K2 ),STDD ( K2 )
 200  CONTINUE
      IF ( OPT .NE. 1 ) GO TO 210
           WRITE ( 6,1160 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
           GO TO 220
 210  CONTINUE
      WRITE ( 6,1140 ) (STNAME(3,K1),K1=1,5),ANGD(COUNTA),ANGM(COUNTA),
     *ANGS(COUNTA),STDA(COUNTA)
      WRITE ( 6,1130 ) ( STNAME ( 4,K1 ) , K1 = 1,5 )
 220  CONTINUE
      WRITE ( 6,1000 )
C
C
C     END PRINT_INPUT_DATA
C
C     ALGORITHM COMPUTE_ADJUST_APPROXIMATE
C
C     THE CALCULATION OF OPEN TRAVERSE
C
C
      CALL CDMSR ( NUM360,NUMO,NUMO,AN360R )
      ITER1 = 2
      IF ( OPT .NE. 1 ) GO TO 230
           ITER1 = 1
 230  CONTINUE
      DO 390 ADJ1 = 1,ITER1
         XGD = EGRID ( 2 )
         YGD = NGRID ( 2 )
         AZFIR = AZ21
         DO  250  I = 1,COUNTD
             AZIMUT = AZFIR + ANG ( I )
             IF ( AZIMUT .LT. AN360R ) GO TO 240
                  AZIMUT = AZIMUT - AN360R
 240         CONTINUE
             COAZR ( I ) = AZIMUT
             CALL CROMS ( AZIMUT,DUM1D,DUM1M,DUM1S )
             CFAZD ( I ) = IDINT ( DUM1D )
             CFAZM ( I ) = IDINT ( DUM1M )
             CFAZS ( I ) = DUM1S
             DISTAN = DIST ( I )
```

```
            CALL CXYC ( XGD,YGD,DISTAN,AZIMUT,DUMMY1,DUMMY2 )
            DELTAX ( I ) = DUMMY1 - XGD
            DELTAY ( I ) = DUMMY2 - YGD
            SUMDX = SUMDX + DELTAX ( I )
            SUMDY = SUMDY + DELTAY ( I )
            CEGRID ( I ) = DUMMY1
            CNGRID ( I ) = DUMMY2
            CALL CTANA ( DUMMY1,DUMMY2,XGD,YGD,AZC,DUMIS )
            CALL CRDMS ( AZC,DUMID,DUMIM,DUMIS )
            CBAZD ( I ) = IDINT ( DUMID )
            CBAZM ( I ) = IDINT ( DUMIM )
            CBAZS ( I ) = DUMIS
            AZFIR = AZC
            XGD = DUMMY1
            YGD = DUMMY2
  250     CONTINUE
C
C
C      PRINT DETAILS OF PRECALCULATED OF TRAVERSE STATION POSITIONS
C
C
          IF ( OPT .EQ. 1 ) GO TO 280
               IF ( ADJ1 .NE. 1 ) GO TO 260
                    WRITE ( 6,1180 )
                    WRITE ( 6,1190 )
                    GO TO 290
  260          CONTINUE
               CORDX1 = ( SUMDX - ( EGRID (3) - EGRID (2) )) / TODIS
               CORDY1 = ( SUMDY - ( NGRID (3) - NGRID (2) )) / TODIS
               XGD = EGRID ( 2 )
               YGD = NGRID ( 2 )
               DO 270 I = 1,COUNTD
                  CEGRID ( I ) = XGD + ( DELTAX(I) - ( CORDX1*DIST(I)))
                  CNGRID ( I ) = YGD + ( DELTAY(I) - ( CORDY1*DIST(I)))
                  XGD = CEGRID ( I )
                  YGD = CNGRID ( I )
  270          CONTINUE
               WRITE ( 6,1000 )
               WRITE ( 6,1290 )
               WRITE ( 6,1300 )
               GO TO 290
  280     CONTINUE
          WRITE ( 6,1170 )
          WRITE ( 6,1200 )
  290     CONTINUE
          WRITE ( 6,1030 )
          WRITE ( 6,1040 )
          WRITE ( 6,1050 )
```

```
      WRITE ( 6,1060 )
      WRITE ( 6,1210 ) (STNAME(2,K1),K1=1,5),NGRID(2),EGRID(2)
      DO  330 I = 1,COUNTD
          K4 = I
          K5 = I + 1
          WRITE ( 6,1220 ) K4,K5,CFAZD(I),CFAZM(I),CFAZS(I)
          IF ( OPT .EQ. 1 ) GO TO 310
               IF ( I .NE. COUNTD ) GO TO 300
                  WRITE ( 6,1070 ) K5,(STNAME(3,K1),K1=1,5),CNGRID(I
     *            ),CEGRID(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
  300          CONTINUE
  310     CONTINUE
          IF ( I .EQ. COUNTD ) GO TO 320
               WRITE ( 6,1070 ) K5,(TRNAME(I,K1),K1=1,5),CNGRID(I),
     *         CEGRID(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
  320     CONTINUE
  330 CONTINUE
      IF ( OPT .NE. 1 ) GO TO 340
          WRITE ( 6,1070 ) K5,(TRNAME(COUNTD,K1),K1=1,5),CNGRID(COU
     *    NTD),CEGRID(COUNTD),K5,K4,CBAZD(COUNTD),CBAZM(COUNTD),CBA
     *    ZS(COUNTD)
  340 CONTINUE
      IF ( OPT .EQ. 1 ) GO TO 380
          AZLAS = AZFIR + ANG ( COUNT )
          IF ( AZLAS .LT. AN360R ) GO TO 350
               AZLAS = AZLAS - AN360R
  350     CONTINUE
          K4 = I
          K5 = I + 1
          CALL CRDMS ( AZLAS,DUM1D,DUM1M,DUM1S )
          TEMD = IDINT ( DUM1D )
          TEMM = IDINT ( DUM1M )
          WRITE ( 6,1220 ) K4,K5,TEMD,TEMM,DUM1S
          WRITE ( 6,1230 ) K5,(STNAME(4,K1),K1=1,5)
          DIFAZI = AZLAS - AZ34
          CORAZ = DIFAZI / DFLOAT ( COUNTA )
          CALL CRDMS ( DIFAZI,DUM1D,DUM1M,DUM1S )
          EAZ1D = IDINT ( DABS ( DUM1D ))
          EAZ1M = IDINT ( DABS ( DUM1M ))
          EAZ1S = DABS ( DUM1S )
          WPRED ( 1 ) = DIFAZI
          CALL CRDMS ( CORAZ,DUM1D,DUM1M,DUM1S )
          EAZ2D = IDINT ( DABS ( DUM1D ))
          EAZ2M = IDINT ( DABS ( DUM1M ))
          EAZ2S = DABS ( DUM1S )
          WPRED ( 2 ) = SUMDX - ( EGRID(3) - EGRID(2) )
          WPRED ( 3 ) = SUMDY - ( NGRID(3) - NGRID(2) )
          DIFDIS = DSQRT ( WPRED(2)**2+WPRED(3)**2 )
```

```
                ERR = IDINT ( TODIS / DIFDIS )
                IF ( ADJ1 .NE. 1 ) GO TO 370
                     WRITE ( 6,1240 ) EAZ1D,EAZ1M,EAZ1S
                     WRITE ( 6,1250 ) EAZ2D,EAZ2M,EAZ2S
                     WRITE ( 6,1260 ) TODIS
                     WRITE ( 6,1270 ) DIFDIS
                     WRITE ( 6,1280 ) ERR
C
C
C     CORRECTED OBSERVED ANGLES
C
C
                     DO 360 I = 1,COUNTA
                        ANG ( I ) = ANG ( I ) - CORAZ
 360                 CONTINUE
                     SUMDX = 0.0D0
                     SUMDY = 0.0D0
 370            CONTINUE
 380        CONTINUE
 390  CONTINUE
      IF ( OPT .EQ. 1 ) GO TO 400
           WRITE ( 6,1310 )
           WRITE ( 6,1320 ) EAZ1D,EAZ1M,EAZ1S
           SUMDX = EGRID ( 3 ) - CEGRID ( COUNTD )
           SUMDY = NGRID ( 3 ) - CNGRID ( COUNTD )
           DIFDIS = DSQRT ( SUMDX**2 + SUMDY**2 )
          WRITE ( 6,1330 ) DIFDIS
 400  CONTINUE
      WRITE ( 6,1000 )
C
C
C     END COMPUTE_ADJUST_APPROXIMATE
C
C
      STOP
C
1000 FORMAT ( '1' )
1010 FORMAT ( ///,23X,'KNOWN HORIZONTAL STATION POSITIONS' )
1020 FORMAT (      23X,'**********************************', // )
1030 FORMAT ( 31X,'UTM GRID COORDINATES',6X,'AZIMUTH   GRID AZIMUTH',/)
1040 FORMAT ( 28X,'GRID NORTHING  GRID EASTING  FROM TO',4X,'CLOCKWISE'
    *,/ )
1050 FORMAT ( 1X,'NO',3X,'NAME OF STATIONS',35X,'NO',3X,'NO',4X,'FROM N
    *ORTH', / )
1060 FORMAT ( 32X,'METERS',8X,'METERS',14X,'D. M.  S.',/ )
1070 FORMAT ( 1X,I2,1X,5A4,3X,F14.5,2X,F12.5,2X,I2,' ->',I2,1X,I3,1X,I2
    *,1X,F8.5,/ )
1080 FORMAT ( ///,31X,'OBSERVED DATA' )
```

```
1090 FORMAT (       31X,'*************',// )
1100 FORMAT ( 29X,'OBSERVED',19X,'  GRID  ',/ )
1110 FORMAT ( 6X,'NAME OF STATIONS',8X,'ANGLE',8X,'STD.',9X,'DISTANCE',
    *5X,'STD.',/ )
1120 FORMAT ( 26X,'D. M.  S.', 9X,'S.',11X,'METERS',7X,'M.' )
1130 FORMAT ( /,4X,5A4,/ )
1140 FORMAT ( 4X,5A4,1X,I3,1X,I2,1X,F8.5,1X,F8.5 )
1150 FORMAT ( 52X,F14.5,1X,F8.5 )
1160 FORMAT ( 4X,5A4,/ )
1170 FORMAT ( ///,17X,'CALCULATION OF OPEN TRAVERSE STATION POSITIONS')
1180 FORMAT ( ///,15X,'PRECALCULATED OF CLOSED TRAVERSE STATION POSITIO
    *NS' )
1190 FORMAT ( 15X,'************************************************',
    */// )
1200 FORMAT ( 17X,'**********************************************',// )
1210 FORMAT ( 2X,'1',1X,5A4,3X,F14.5,2X,F12.5,/ )
1220 FORMAT ( 57X,I2,' ->',I2,1X,I3,1X,I2,1X,F8.5,/ )
1230 FORMAT ( 1X,I2,1X,5A4,/ )
1240 FORMAT ( ///,2X,'TOTAL AZIMUTH ERROR',8X,'= ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1250 FORMAT ( /,2X,'AZIMUTH ERROR PER STATIONS = ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1260 FORMAT ( /,'  TOTAL GRID DISTANCE         = ',F16.5,' METERS.' )
1270 FORMAT ( /,2X,'DISTANCE ERROR',13X,'= ',F10.5,' METERS. ' )
1280 FORMAT ( /,2X,'ACCURACY',18X,' =  1:',I12 )
1290 FORMAT ( ///,6X,'ADJUSTMENT OF CLOSED TRAVERSE STATION POSITIONS B
    *Y APPROXIMATE METHOD' )
1300 FORMAT (       6X,'*************************************************
    *********************',// )
1310 FORMAT ( ///,2X,'CHECK ROUND OFF ERROR ' )
1320 FORMAT ( /,2X,'AZIMUTH   = ',I3,' D. ',I2,' M. ',F8.5,' S. ' )
1330 FORMAT ( /,2X,'DISTANCE  = ',F10.5,' METERS. ' )
1340 FORMAT ( I1 )
1350 FORMAT ( I2 )
1360 FORMAT ( F20.12 )
1370 FORMAT ( 5A4 )
1380 FORMAT ( 10X,'**************************************************
    ********' )
1390 FORMAT ( 10X,'*',57X,'*' )
1400 FORMAT ( 10X,'*',9X,'THIS IS A PROGRAM TO CALCULATION OF TRAVERSE'
    *,4X,'*'/,10X,'*',4X,'STATION POSITIONS. THERE ARE TWO OPTIONS IN T
    *HIS',5X,'*',/,10X,'*',4X,'PROGRAM.:-',43X,'*' )
1410 FORMAT ( 10X,'*',4X,'1 CALCULATION OF OPEN TRAVERSE STATION POSITI
    *ONS.',4X,'*' )
1420 FORMAT ( 10X,'*',4X,'2 CALCULATION OF CLOSED TRAVERSE STATION POSI
    *TIONS',3X,'*' )
1430 FORMAT (  /,28X,'WHAT IS YOUR OPTION ?' )
1440 FORMAT (   ,20X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/    )
```

```
1450 FORMAT ( //,16X,'THERE ARE TWO OPTIONS TO ENTER DATA.',/ )
1460 FORMAT (     14X,'1 ENTER DATA BY DATA FILE.',/ )
1470 FORMAT (14X,'2 ENTER DATA BY INTERACTIVE FROM TERMINAL KEYBOARD.')
1480 FORMAT ( /,21X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/ )
1490 FORMAT ( //,16X,'WHAT IS THE NAME OF KNOWN POSITION # ',I1,'.?')
1500 FORMAT ( /,12X,'THIS KNOWN POSITION WAS USED FOR THE FIRST BACKSIG
    *HT.')
1510 FORMAT ( //,12X,'THIS KNOWN POSITION WAS USED FOR THE LAST FORESIG
    *HT.' )
1520 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE FIRST POSITION AT WHICH
    *',/,12X,'AN ANGLE WAS OBSERVED.' )
1530 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE LAST POSITION AT WHICH'
    *,/,12X,'AN ANGLE WAS OBSERVED.' )
1540 FORMAT ( /,12X,'FOR EXAMPLE : GRID NORTHING 4,072,555.85206 METERS
    *.',/,26X,'ENTER 4072555.85206D0' )
1550 FORMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION OF THE OBSERVED AN
    *GLE',/,12X,'AT THIS STATION IN SECONDS ?',/,12X,'FOR EXAMPLE STD.
    *= 3.9 SECONDS. : ENTER 3.90D0 ' )
1560 FORMAT ( /,12X,'FOR EXAMPLE : GRID EASTING 608,279.04404 METERS.',
    */,26X,'ENTER 608279.04404D0' )
1570 FORMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION IN METERS.?',/,12X
    *,'FOR EXAMPLE  STD. = .011 METERS : ENTER 0.011D0' )
1580 FORMAT ( /,12X,'HOW MANY POSITIONS DID YOU OBSERVE ANGLES ?',//,19
    *X,'PLEASE ENTER 01 TO 32' )
1590 FORMAT ( /,12X,'ENTER THE OBSERVED ANGLE AT THE FIRST KNOWN POSITI
    *ON.' )
1600 FORMAT ( /,10X,'ENTER THE OBSERVED ANGLE AT TRAVERSE STATION POSIT
    *ION # ',I2,'.' )
1610 FORMAT ( /,10X,'FOR EXAMPLE : OBSERVED ANGLE CLOCKWISE FROM BACKSI
    *GHT TO FORESIGHT',/,24X,'IS 169 D. 32 M. 11.11 S.',//,24X,'ENTER 1
    *69.32111110D0' )
1620 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN THE FIRST KNOWN POSITI
    *ON AND',/,12X,'TRAVERSE STATION POSITION # 1.' )
1630 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN TRAVERSE STATION POSIT
    *ION # ',I2,'.',/,12X,'AND TRAVERSE STATION POSITION # ',I2,'.' )
1640 FORMAT ( /,12X,'FOR EXAMPLE : THE GRID DISTANCE BETWEEN THE STATIO
    *NS IS',/,26X,'399.052 METERS.',//,26X,'ENTER 399.05200D0' )
1650 FORMAT ( //,12X,'WHAT IS THE NAME OF TRAVERSE STATION POSITION # '
    *,I2,'.' )
1660 FORMAT ( //,12X,'WHAT IS UTM GRID NORTHING AT THIS POSITION ? ' )
1670 FORMAT ( //,12X,'WHAT IS UTM GRID EASTING AT THIS POSITION ? ' )
1680 FORMAT ( //,16X,'WHAT IS UTM GRID AZIMUTH FROM',/,12X,5A4,' TO  ',
    *5A4 )
1690 FORMAT ( //,12X,'FOR EXAMPLE : UTM GRID AZIMUTH FROM  A.  TO  B.',
    */,26X,'IS 139 D. 19 M. 29.3 S. : ENTER 139.192930D0 ' )
C
C
      END
```

```
C                            TD WATFIV A1
$EJECT
         SUBROUTINE TDMS ( DUMMY1,DEG1,MIN1,SEC1 )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C        AUTHOR : LT. SAMAN AUMCHANTR RTN                            C
C          DATE : OCTOBER 23, 1983                                   C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    P
C         THIS IS A SUBROUTINE FOR COMPUTING DEGREE , MINUTE  AND SECOND P
C     FROM INPUT DATA OF THE FORM DDD.MMSSSSSSS                      P
C     INPUT   : DUMMY1                                               P
C     OUTPUT  : DEG1 , MIN1 AND SEC1                                 P
C                                                                    P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                    V
C                 VARIABLE DEFINITIONS                               V
C                                                                    V
C          DUMMY1 = INPUT DATA IN DEGREE . MINUTE SECOND             V
C          DUMMY2 = INPUT DATA IN MINUTE . SECOND                    V
C          DEG1   = INPUT DATA IN DEGREE                             V
C          MIN1   = INPUT DATA IN MINUTE                             V
C          SEC1   = INPUT DATA IN SECOND                             V
C          SIG    = THE SIGN OF THE INPUT DATA                       V
C          NUM5   = 100000                                           V
C          NUM7   = 10000000                                         V
C          NUM100 = 100                                              V
C                                                                    V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                    C
C                 VARIABLE DECLARATIONS                              C
C
       DOUBLE PRECISION DUMMY1,DUMMY2,DEG1,MIN1,SEC1,NUM100,NUM7,NUM5,SIG
C
       DATA NUM100/100.0D0/,NUM7/10000000.0D0/,NUM5/100000.0D0/
C
C      CHANGE THE INPUT DATA TO DEGREE , MINUTE AND SECOND
C
       SIG = ( DUMMY1 / DABS ( DUMMY1 ))*0.50D0
       DEG1   = DFLOAT ( IDINT ( DUMMY1 ))
       DUMMY2 = ( DUMMY1 - DEG1 ) * NUM100
       MIN1   = DFLOAT ( IDINT ( DUMMY2 ))
       SEC1   = ( DFLOAT ( IDINT ((( DUMMY2 - MIN1 )*NUM7 )+SIG))) / NUM5
C
       RETURN
       END
```

```
C
C                           CD WATFIV A1
$EJECT
C
      SUBROUTINE CDMSR ( D1,M1,S1,R1 )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN.                            C
C        DATE : OCTOBER 23, 1983                                    C
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   P
C           THIS IS A SUBROUTINE FOR CHANGE THE ANGLES FROM DEGREE ,P
C       MINUTE AND SECOND TO RADIAN.                                P
C       INPUT    : D1 , M1 AND S1                                   P
C       OUTPUT   : R1                                               P
C                                                                   P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                   V
C                 VARIABLE DEFINITIONS                              V
C                                                                   V
C         PI1    = THE VALUE OF PI                                  V
C         D1     = THE VALUE OF THE ANGLE  IN DEGREE                V
C         M1     = THE VALUE OF THE ANGLE  IN MINUTE                V
C         S1     = THE VALUE OF THE ANGLE  IN SECOND                V
C         R1     = THE VALUE OF THE ANGLE  IN RADIAN                V
C         N60    = 60                                               V
C         N180   = 180                                              V
C         N3600  = 3600                                             V
C                                                                   V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                   C
C        VARIABLE DECLARATIONS                                      C
C                                                                   C
      DOUBLE PRECISION D1,M1,S1,R1,PI1,N60,N3600,N180
C
C
      DATA PI1/3.141592653589793D0/,N60/60.0D0/,N3600/3600.0D0/,N180/180
     *.0D0/
C
C
      R1 = (( D1 + ( M1/N60 ) + ( S1/N3600 ))*PI1 ) / N180
C
C
      RETURN
C
      END
```

```
C                     CR WATFIV A1
$EJECT
C
      SUBROUTINE CRDMS ( RR1,D2,M2,S2 )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     AUTHOR : LT. SAMAN AUMCHANTR.                                    C
C       DATE : OCTOBER 23, 1983                                        C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      P
C            THIS IS A SUBROUTINE FOR CHANGE THE ANGLES FROM RADIAN    P
C        TO DEGREE , MINUTE AND SECOND.                                P
C        INPUT  : RR1                                                  P
C        OUTPUT : D2 , M2 AND S2                                       P
C                                                                      P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                      V
C                  VARIABLE DEFINITIONS                                V
C                                                                      V
C        PI1     = THE VALUE OF PI                                     V
C        D2      = THE VALUE OF THE ANGLE IN DEGREE                    V
C        M2      = THE VALUE OF THE ANGLE IN MINUTE                    V
C        S2      = THE VALUE OF THE ANGLE IN SECOND                    V
C        RR1     = THE VALUE OF THE ANGLE IN RADIAN                    V
C        TDEG    = THE TOTAL VALUE OF THE ANGLE IN DEGREE              V
C        TMIN    = THE TOTAL VALUE OF THE ANGLE IN MINUTE              V
C        DIF     = THE DIFFERENCE OF THE VALUE BETWEEN REAL AND INTEGER V
C                                                                      V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                      C
C         VARIABLE DECLARATIONS                                        C
C                                                                      C
      DOUBLE PRECISION RR1,D2,M2,S2,PI1,N60,N180,TDEG,DIF,TMIN
      DATA PI1/3.14159265358979300/,N60/60.0D0/,N180/180.0D0/
      TDEG = ( N180*RR1 ) / PI1
      D2 = DFLOAT ( IDINT ( TDEG ))
      DIF  = TDEG - D2
      TMIN = DIF*N60
      M2 = DFLOAT ( IDINT ( TMIN ))
      DIF  = TMIN - M2
      S2   = DIF*N60
C
      RETURN
C
      END
C
```

```
C                               CP WATFIV A1
$EJECT
      SUBROUTINE CXYC ( X1,Y1,DIS12,ANG12R,X2,Y2 )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C     AUTHOR : LT. SAMAN AUMCHANTR RTN                               C
C       DATE : FEBRUARY 11 , 1984                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    P
C       THIS IS A SUBROUTINE FOR COMPUTATION OF UTM GRID COORDINATES P
C    INPUT  : X1 , Y1 , DIST12 AND ANG12R                            P
C    OUTPUT : X2 AND Y2                                              P
C                                                                    P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                    V
C                VARIABLE DEFINITIONS                                V
C                                                                    V
C      X1      = THE GRID EASTING OF THE KNOWN POSITION              V
C      Y1      = THE GRID NORTHING OF THE KNOWN POSITION             V
C      X2      = THE GRID EASTING OF THE UNKNOWN POSITION            V
C      Y2      = THE GRID NORTHING OF THE UNKNOWN POSITION           V
C      DIS12   = THE DISTANCE BETWEEN THE KNOWN POSITION AND THE     V
C                UNKNOWN POSITION                                    V
C      ANG12R  = THE GRID AZIMUTH CLOCKWISE FROM THE NORTH IN RADIAN V
C      DIFX    = THE DIFFERENCE IN THE DISTANCE IN X - AXIS          V
C      DIFY    = THE DIFFERENCE IN THE DISTANCE IN Y - AXIS          V
C                                                                    V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                    C
C                VARIABLE DECLARATIONS                               C
C                                                                    C
      DOUBLE PRECISION X1,Y1,X2,Y2,DIS12,ANG12R,DIFX,DIFY
C
C     THE CALCULATION OF THE DIFFERENCE IN DISTANCES IN X AND Y-AXES
C
      DIFX = DIS12*( DSIN ( ANG12R ) )
      DIFY = DIS12*( DCOS ( ANG12R ) )
C
C     THE CALCULATION OF THE UNKNOWN POSITION
C
      X2   = X1 + DIFX
      Y2   = Y1 + DIFY
C
      RETURN
C
      END
```

```
C                                  CT WATFIV A1
$EJECT
C
      SUBROUTINE CTANA ( X1,Y1,X2,Y2,ANGLR,DIST12 )
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN                               C
C        DATE : JANUARY 11 , 1984                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     P
C       THIS IS A SUBROUTINE FOR CALCULATION OF THE GRID AZIMUTH AND  P
C   THE DISTANCE BETWEEN THE KNOWN POSITIONS # 1 AND # 2              P
C   INPUT  : X1 , Y1 , X2 AND Y2                                      P
C   OUTPUT : ANGLR AND DIST12                                         P
C                                                                     P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                     V
C                 VARIABLE DEFINITIONS                                V
C                                                                     V
C   X1,X2      = THE GRID EASTING OF THE KNOWN POSITIONS              V
C   Y1,Y2      = THE GRID NORTHING OF THE KNOWN POSITIONS             V
C   ANGLR      = THE GRID AZIMUTH CLOCKWISE FROM THE NORTH IN RADIAN  V
C   DIST12     = THE DISTANCE BETWEEN THE  KNOWN POSITIONS # 1 AND # 2 V
C                IN METERS                                            V
C   DIFX       = THE DIFFERENCE IN THE DISTANCE IN X - AXIS           V
C   DIFY       = THE DIFFERENCE IN THE DISTANCE IN Y - AXIS           V
C                                                                     V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                     C
C                 VARIABLE DECLARATIONS                               C
C
      DOUBLE PRECISION X1,Y1,X2,Y2,ANGLR,DIFX,DIFY,NUM0,NUM90,NUM180,NUM
     *270,NU90R,NU180R,NU270R,DIST12
C
C
      DATA NUM0/0.0D0/,NUM90/90.0D0/,NUM180/180.0D0/,NUM270/270.0D0/
C
C
C     CALL SUBROUTINE CDMSR TO CHANGE THE ANGLES FROM DEGREE , MINUTE
C                              AND SECOND TO RADIAN
C
C
      CALL CDMSR ( NUM90,NUM0,NUM0,NU90R )
      CALL CDMSR ( NUM180,NUM0,NUM0,NU180R )
      CALL CDMSR ( NUM270,NUM0,NUM0,NU270R )
```

```
C
C
C     THE CALCULATION OF THE DIFFERENCE IN DISTANCES IN X AND Y-AXES
C
C
      DIFX = X2 - X1
      DIFY = Y2 - Y1
C
C
C     THE CALCULATION OF THE GRID AZIMUTH
C
C
      IF ( DIFX .NE. NUMO .OR. DIFY .NE. NUMO ) GO TO 10
           ANGLR = NUMO
           GO TO 50
   10 CONTINUE
      IF ( DIFX .NE. NUMO ) GO TO 30
           IF ( DIFY .LT. NUMO ) GO TO 20
                ANGLR = NUMO
                GC TO 50
   20      CONTINUE
           ANGLR = NU180R
           GO TO 50
   30 CONTINUE
      IF ( DIFX .LT. NUMO ) GO TO 40
           ANGLR = NU90R - DATAN ( DIFY / DIFX )
           GO TO 50
   40 CONTINUE
      ANGLR = NU270R - DATAN ( DIFY / DIFX )
   50 CONTINUE
C
C
C     THE CALCULATION OF THE DISTANCE BETWEEN THE KNOWN POSITIONS
C                           # 1 AND # 2
C
C
      DIST12 = DSQRT ( DIFX**2 + DIFY**2 )
C
C
C
      RETURN
C
C
C
      END
C
C
```

APPENDIX B

PROGRAM 2

COMPUTER OUTPUT

THE CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
BY INDIRECT OBSERVATIONS METHOD
FOR UNCORRELATED OBSERVATIONS WITH EQUAL PRECISION

KNOWN HORIZONTAL STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM NO TO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	PIPHER				
2	MOSS 2	4072555.85206	608279.04404	2 -> 1	100 16 23.77800
3	HOLM	4079258.31754	612238.85256	3 -> 4	136 33 26.33400
4	MORAN				

OBSERVED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	STD. S.	GRID DISTANCE METERS	STD. M.
PIPHER				
MOSS 2	246 5 43.20000	1.98400		
			1424.00400	0.00100
MOSSBACK	222 51 8.60000	1.40500		
			365.74400	0.00100
DUNE TEMP	190 15 2.60000	1.20300		
			6476.27100	0.00300
HOLM	277 5 17.00000	1.61400		
MORAN				

PRECALCULATED OF CLOSED TRAVERSE STATION POSITIONS
****

NC	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 6.97800
2	MOSSBACK	4073939.74473	607943.44238	2 -> 1	166 22 6.97800
				2 -> 3	29 13 15.57800
3	DUNE TEMP	4074258.94534	608121.99110	3 -> 2	209 13 15.57800
				3 -> 4	39 28 18.17800
4	HOLM	4079258.22818	612238.93865	4 -> 3	219 28 18.17800
				4 -> 5	136 33 35.17800
5	MORAN				

TOTAL AZIMUTH ERROR = 0 D. 0 M. 8.84400 S.
AZIMUTH ERROR PER STATIONS = 0 D. 0 M. 2.21100 S.
TOTAL GRID DISTANCE = 8266.01900 METERS.
DISTANCE ERROR = 0.12408 METERS.
ACCURACY = 1: 666 17

CALCULATION OF CLOSED TRAVERSE STATION POSITIONS BY LEAST-SQUARES ADJUSTMENT
****

BY INDIRECT OBSERVATIONS METHOD

		UTM GRID COORDINATES			
		GRID NORTHING	STD.	GRID EASTING	STD.
NO	NAME OF STATIONS	METERS	M.	METERS	M.
1	MOSSBACK	4073939.74899	0.00026	607943.45994	0.00107
2	DUNE TEMP	4074258.95013	0.00267	608122.00770	0.00324

THE STANDARD DEVIATION OF UNIT WEIGHT IS 0.0000201144

THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE

ANGLE # 1 = 0.1591744830 SECONDS

ANGLE # 2 = 2.9427035466 SECONDS

ANGLE # 3 = 2.9173152616 SECONDS

ANGLE # 4 = 0.1336986660 SECONDS

THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE

DISTANCE # 1 = 0.0000201144 METERS

DISTANCE # 2 = 0.0000201144 METERS

DISTANCE # 3 = 0.0000201144 METERS

THE CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
BY INDIRECT OBSERVATIONS METHOD
FOR UNCORRELATED OBSERVATIONS WITH UNEQUAL PRECISION

KNOWN HORIZONTAL STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	PIPHER				
2	MOSS 2	4072555.85206	608279.04404	2 -> 1	100 16 23.77800
3	HOLM	4079258.31754	612238.85256	3 -> 4	136 33 26.33400
4	MORAN				

OBSERVED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	STD. S.	GRID DISTANCE METERS	STD. M.
PIPHER				
MOSS 2	246 5 43.20000	1.98400		
			1424.00400	0.00100
MOSSBACK	222 51 8.60000	1.40500		
			365.74400	0.00100
DUNE TEMP	190 15 2.60000	1.20300		
			6476.27100	0.00300
HOLM	277 5 17.00000	1.61400		
MORAN				

PRECALCULATED OF CLOSED TRAVERSE STATION POSITIONS
****

NC	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	UTM GRID COORDINATES GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 6.97800
2	MOSSBACK	4073939.74473	607943.44238	2 -> 1	166 22 6.97800
				2 -> 3	29 13 15.57800
3	DUNE TEMP	4074258.94534	608121.99110	3 -> 2	209 13 15.57800
				3 -> 4	39 28 18.17800
4	HOLM	4079258.22818	612238.93865	4 -> 3	219 28 18.17800
				4 -> 5	136 33 35.17800
5	MORAN				

TOTAL AZIMUTH ERROR = 0 D. 0 M. 8.84400 S.

AZIMUTH ERROR PER STATIONS = 0 D. 0 M. 2.21100 S.

TOTAL GRID DISTANCE = 8266.01900 METERS.

DISTANCE ERROR = 0.12408 METERS.

ACCURACY = 1: 66617

CALCULATION OF CLOSED TRAVERSE STATION POSITIONS BY LEAST-SQUARES ADJUSTMENT
****

BY INDIRECT OBSERVATIONS METHOD

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	STD. M.	GRID EASTING METERS	STD. M.
1	MOSSBACK	4073939.74809	0.00317	607943.45522	0.00954
2	DUNE TEMP	4074258.95029	0.00405	608122.00189	0.01032

THE STANDARD DEVIATION OF UNIT WEIGHT IS 2.6968502720

THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE

ANGLE # 1 = 1.4034554964 SECONDS

ANGLE # 2 = 2.6410292243 SECONDS

ANGLE # 3 = 2.5525970615 SECONDS

ANGLE # 4 = 0.2613735339 SECONDS

THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE

DISTANCE # 1 = 0.0026856614 METERS

DISTANCE # 2 = 0.0026737945 METERS

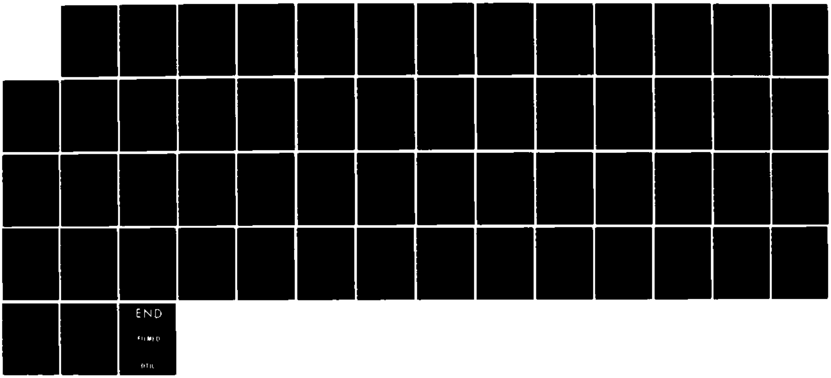
DISTANCE # 3 = 0.0074504772 METERS

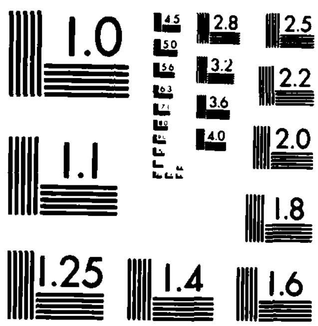
PROGRAM LISTING

```
C               TRAVIND WATFIV A1
C
$JOB            SAM2,NOCHECK
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN.                              C
C        DATE : AUG 11 , 1984                                         C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     P
C        THIS IS A PROGRAM FOR CALCULATION  OF TRAVERSE STATION       P
C    POSITIONS BY INDIRECT OBSERVATIONS METHOD.                       P
C                                                                     P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                     V
C                 VARIABLE DEFINITIONS                                V
C                                                                     V
C       OPT     = OPTION FOR CALCULATION OF OPEN OR CLOSED TRAVERSE   V
C       CHOIC   = OPTION FOR ENTER DATA BY DATA FILE OR INTERACTIVE   V
C                 FROM TERMINAL KEYBOARD.                             V
C       DUMMY1,DUMMY2,DUM1D,DUM1M,DUM1S = THE DUMMY VARIABLES         V
C       NGRID(4)= GRID NORTHING OF THE KNOWN POSITIONS.               V
C       EGRID(4)= GRID EASTING OF THE KNOWN POSITIONS.                V
C       STNAME(4,5) = THE NAME OF KNOWN STATIONS.                     V
C       TRNAME( 32,5) = THE NAME OF TRAVERSE STATIONS.                V
C       DIRECD(4),DIRECM(4),DIRECS(4) = THE GRID AZIMUTH OF KNOWN     V
C                 POSITIONS.                                          V
C       STAT1(4),STAT2(4) = THE NUMBER OF KNOWN POSITIONS.            V
C       COUNT, COUNTA = THE NUMBER OF OBSERVED ANGLES.                V
C       ANG( 32)= THE OBSERVED ANGLE IN RADIAN.                       V
C       ANGD( 32),ANGM( 32),ANGS( 32) = THE OBSERVED ANGLES IN DEGREE,V
C                 MINUTE AND SECOND.                                  V
C       COUNTN, COUNTD = THE NUMBER OF MEASURED DISTANCE.             V
C       DIST( 32) = THE MEASURED DISTANCE IN METERS.                  V
C       TODIS   = THE SUM OF MEASURED DISTANCE.                       V
C       CFAZD( 32),CFAZM( 32),CFAZS( 32) = THE COMPUTED FORWARD AZIMUTH V
C       CBAZD( 32),CBAZM( 32),CBAZS( 32) = THE COMPUTED BACK AZIMUTH  V
C       CNGRID( 32) = GRID NORTHING OF TRAVERSE STATION POSITIONS.    V
C       CEGRID( 32) = GRID EASTING OF TRAVERSE STATION POSITIONS.     V
C       ERR     = THE TOTAL DISTANCE ERROR.                           V
C       DIFAZI  = THE TOTAL ANGLE ERROR IN RADIAN.                    V
C       EAZID, EAZIM, EAZIS = THE TOTAL ANGLE ERROR IN DEGREE,        V
```

AD-A153 612   A COMPARISON OF METHODS OF LEAST SQUARES ADJUSTMENT OF TRAVERSES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA S AUMCHANTR DEC 84   2/2

UNCLASSIFIED   F/G 12/1   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

```
C             MINUTE AND SECOND.                                        V
C      EAZ2D, EAZ2M, EAZ2S = THE ANGLE ERROR PER STATION.               V
C      TDMS     = SUBROUTINE FOR CHANGE INPUT DATA FORM DDD.MMSSSSSSS   V
C                 TO DEGREE , MINUTE AND SECOND.                        V
C      CDMSR    = SUBROUTINE FOR CHANGE DEGREE, MINUTE AND              V
C                 SECOND TO RADIAN.                                     V
C      CRDMS    = SUBROUTINE FOR CHANGE RADIAN TO DEGREE, MINUTE        V
C                 AND SECOND.                                           V
C      CTANA    = SUBROUTINE FOR CALCULATION OF GRID AZIMUTH AND THE    V
C                 THE DISTANCE BETWEEN POSITIONS # 1 AND #2             V
C      CXYC     = SUBROUTINE FOR COMPUTATION OF THE UTM GRID COORDINATES V
C      LSTR     = SUBROUTINE TO ADJUSTED THE COORDINATES OF CLOSED      V
C                 TRAVERSE BY INDIRECT OBSERVATIONS METHOD              V
C                                                                       V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                       C
C                                                                       C
C                     VARIABLE DECLARATIONS                             C
C                                                                       C
C                                                                       C
       DOUBLE PRECISION DUMMY1,CFAZS(32),CBAZS(32),EAZ1S,EAZ2S,CEGRID(32)
     *,CNGRID(32),DIS21,AZ21,AZ12,DIS34,AZ34,DUM1D,AZ43,DUM1M,DUM1S,DU
     *MMY2,ANG(32),DIST(32),AZFIR,AZIMUT,NUMO,NUM360,AN360R,DISTAN,AZC,
     *DIFDIS,AZLAS,DIFAZI,CORAZ,TODIS,NUM1,ASNY(32),ASEX(32),COAZR(32)
       DOUBLE PRECISION ANGS(32),DIRECS(4),NGRID(4),EGRID(4),WPRED(3),
     *XGD,YGD,STDA(32),STDD(32),DELTAX(32),DELTAY(32),SUMDX,SUMDY
       DOUBLE PRECISION WM(64,64),STDD(64),FM(64),BM(64,64),BMT(64,64),
     *BMTWM(64,64),NM(64,64),TM(64),DELTA(64),NMI(64,64),WK10(4288),
     *VM(1,64),STDAD
C
C
       INTEGER DES,OPT,COUNT,I,COUNTA,COUNTD,ERR,CHOIC,LOOP1,DIRECD(4),DI
     *RECM(4),STAT1(4),STAT2(4),STNAME(4,5),K1,K2,ANGD(32),ANGM(32),COUN
     *TN,TRNAME(32,5),K4,K5,COUNTP,EAZ1D,EAZ1M,EAZ2D,EAZ2M,CFAZD(32),CBA
     *ZD(32),CBAZM(32),TEMD,TEMM,CFAZM(32),J,NT,ASP,PW,PRINTD
C
C
       DATA NUMO/0.0D0/,NUM360/360.0D0/,TODIS/0.0D0/,NUM1/1.0D0/,SUMDY/0
     *.0D0/,SUMDX/0.0D0/
C
C
C      IF PW = 1; IT MEANS UNWEIGHT MATRIX
C      IF PRINTD = 0; IT MEANS NO PRINT DETAILS OF MATRIX COMPUTATION
C
C
       DATA PW/0/,PRINTD/0/
C
C
```

```
C     ALGORITHM INPUT_DATA
C
C     PRINT TITLE
C
C
      WRITE ( 6,1000 )
      WRITE ( 6,1380 )
      WRITE ( 6,1390 )
      WRITE ( 6,1400 )
      WRITE ( 6,1410 )
      WRITE ( 6,1420 )
      WRITE ( 6,1390 )
      WRITE ( 6,1380 )
      WRITE ( 6,1430 )
      WRITE ( 6,1440 )
C
C
C
C     READ OPTIONS  1 = OPEN TRAVERSE
C                   2 = CLOSED TRAVERSE
C
C
      READ ( 4,1340 ) OPT
      WRITE ( 6,1450 )
      WRITE ( 6,1460 )
      WRITE ( 6,1470 )
      WRITE ( 6,1430 )
      WRITE ( 6,1480 )
C
C
C     READ OPTIONS  1 = ENTER DATA BY DATA FILE
C                   2 = ENTER DATA BY INTERACTIVE
C
C
      READ ( 4,1340 ) CHOIC
C
C
C
C     GET UTM GRID NORTHING, EASTING, AZIMUTH, AND THE NAME
C                   OF KNOWN POSITIONS
C
C
      DES = 04
      IF ( CHOIC .EQ. 2 ) GO TO 10
           DES = 03
 10   CONTINUE
      LOOP1 = 2
      IF ( OPT .EQ. 1 ) GO TO 20
```

```
             LOOP1 = 4
   20     CONTINUE
          DO  70  I = 1,LOOP1
              WRITE ( 6,1490 ) I
              IF ( I .NE. 1 ) GO TO 30
                   WRITE ( 6,1500 )
                   READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
                   GO TO 60
   30         CONTINUE
              IF ( I .NE. 2 ) GO TO 40
                   WRITE ( 6,1520 )
                   READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
                   WRITE ( 6,1670 )
                   WRITE ( 6,1540 )
                   READ ( DES,1360 ) DUMMY1
                   NGRID ( I ) = DUMMY1
                   WRITE ( 6,1680 )
                   WRITE ( 6,1560 )
                   READ ( DES,1360 ) DUMMY1
                   EGRID ( I ) = DUMMY1
                   WRITE ( 6,1690 ) (STNAME(2,K1),K1=1,5),(STNAME(1,K1),K1=1
         *         ,5)
                   WRITE ( 6,1700 )
                   READ ( DES,1360 ) DUMMY1
                   CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
                   DIRECD ( 2 ) = IDINT ( DUM1D )
                   DIRECM ( 2 ) = IDINT ( DUM1M )
                   DIRECS ( 2 ) = DUM1S
                   CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
                   AZ21 = DUMMY1
                   STAT1 ( 2 ) = 2
                   STAT2 ( 2 ) = 1
                   GO TO 60
   40         CONTINUE
              IF ( I .NE. 3 ) GO TO 50
                   WRITE ( 6,1530 )
                   READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
                   WRITE ( 6,1670 )
                   WRITE ( 6,1540 )
                   READ ( DES,1360 ) DUMMY1
                   NGRID ( I ) = DUMMY1
                   WRITE ( 6,1680 )
                   WRITE ( 6,1560 )
                   READ ( DES,1360 ) DUMMY1
                   EGRID ( I ) = DUMMY1
                   GO TO 60
   50         CONTINUE
              IF ( I .NE. 4 ) GO TO 60
```

```
          WRITE ( 6,1510 )
          READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
          WRITE ( 6,1690 ) (STNAME(3,K1),K1=1,5),(STNAME(4,K1),K1=1
     *    ,5)
          WRITE ( 6,1700 )
          READ ( DES,1360 ) DUMMY1
          CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
          DIRECD ( 3 ) = IDINT ( DUM1D )
          DIRECM ( 3 ) = IDINT ( DUM1M )
          DIRECS ( 3 ) = DUM1S
          CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
          AZ34 = DUMMY1
          STAT1 ( 3 ) = 3
          STAT2 ( 3 ) = 4
   60     CONTINUE
   70 CONTINUE
C
C
C     READ THE NUMBER OF OBSERVED ANGLES .
C
C
      WRITE ( 6,1580 )
      READ ( DES,1350 ) COUNT
      COUNTA = COUNT
C
C
C     READ THE OBSERVED ANGLES
C
C
      DO  100 I = 1,COUNTA
          IF ( I .GT. 1 ) GO TO 80
              WRITE ( 6,1590 )
              GO TO 90
   80     CONTINUE
          K2 = I -1
          WRITE ( 6,1600 ) K2
   90     CONTINUE
          WRITE ( 6,1610 )
          READ ( DES,1360 ) DUMMY1
          CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
          CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY2 )
          ANG ( I ) = DUMMY2
          ANGD ( I ) = IDINT ( DUM1D )
          ANGM ( I ) = IDINT ( DUM1M )
          ANGS ( I ) = DUM1S
          WRITE ( 6,1550 )
          READ ( DES,1360 ) DUMMY1
          STDA ( I ) = DUMMY1
```

```
  100 CONTINUE
C
C
C
C     READ THE GRID DISTANCE
C
C
      COUNTD = COLNT
      IF ( OPT .EQ. 1 ) GO TO 110
          COUNTD = COUNT - 1
  110 CONTINUE
      DO  140 I = 1,COUNTD
          IF ( I .GT. 1 ) GO TO 120
              WRITE ( 6,1620 )
              GC TO 130
  120     CONTINUE
          K4 = I - 1
          K5 = I
          WRITE ( 6,1630 ) K4,K5
  130     CONTINUE
          WRITE ( 6,1640 )
          READ ( DES,1360 ) DUMMY1
          DIST ( I ) = DUMMY1
          TODIS = TODIS + DUMMY1
          WRITE ( 6,1570 )
          READ ( CES,1360 ) DUMMY1
          STDD ( I ) = DUMMY1
  140 CONTINUE
      COUNTN = CCUNTD
      IF ( OPT .EQ. 1 ) GO TO 150
          COUNTN = COUNTD - 1
  150 CONTINUE
      DO  160 I = 1,COUNTN
          WRITE ( 6,1650 ) I
          READ ( CES,1370 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
  160 CONTINUE
C
C
C     END INPUT_CATA
C
C
C
C     ALGORITHM PRINT_INPUT_DATA
C
C
C
C     PRINT DETAILS OF THE KNOWN STATION POSITIONS
C
C
      WRITE ( 6,1000 )
```

```
      WRITE ( 6,1010 )
      WRITE ( 6,1020 )
      WRITE ( 6,1030 )
      WRITE ( 6,1040 )
      WRITE ( 6,1050 )
      WRITE ( 6,1060 )
      DO  180 I = 1,LOOP1
          IF ( I .EQ. 2 .OR. I .EQ. 3 ) GO TO 170
              WRITE ( 6,1230 ) I , ( STNAME ( I,K1 ) , K1 = 1,5 )
              GC TO 180
170       CONTINUE
          WRITE ( 6,1070 ) I,(STNAME(I,K1),K1=1,5),NGRID(I),EGRID(I),STA
     *    T1(I),STAT2(I),DIRECD(I),DIRECM(I),DIRECS(I)
180   CONTINUE
      WRITE ( 6,1000 )
      WRITE ( 6,1080 )
      WRITE ( 6,1090 )
      WRITE ( 6,1100 )
      WRITE ( 6,1110 )
      WRITE ( 6,1120 )
      WRITE ( 6,1130 ) ( STNAME ( 1,K1 ) , K1 = 1,5 )
      WRITE ( 6,1140 ) (STNAME(2,K1),K1 = 1,5),ANGD(1),ANGM(1),ANGS(1),S
     *TDA(1)
      WRITE ( 6,1150 ) DIST ( 1 ),STDD ( 1 )
      COUNTP = COUNTD - 1
      IF ( COUNTP .NE. 0 ) GO TO 190
          WRITE ( 6,1160 ) ( TRNAME ( 1,K1 ) , K1 = 1,5 )
          GO TO 220
190   CONTINUE
      DO  200 I = 1,COUNTP
          K2 = I + 1
          WRITE ( 6,1140 ) (TRNAME(I,K1),K1=1,5),ANGD(K2),ANGM(K2),ANGS(
     *    K2),STDA(K2)
          WRITE ( 6,1150 ) DIST ( K2 ),STDD ( K2 )
200   CONTINUE
      IF ( OPT .NE. 1 ) GO TO 210
          WRITE ( 6,1160 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
          GO TO 220
210   CONTINUE
      WRITE ( 6,1140 ) (STNAME(3,K1),K1=1,5),ANGD(COUNTA),ANGM(COUNTA),
     *ANGS(COUNTA),STDA(COUNTA)
      WRITE ( 6,1130 ) ( STNAME ( 4,K1 ) , K1 = 1,5 )
220   CONTINUE
      WRITE ( 6,1000 )
C
C
C
C     END PRINT_INPUT_DATA
```

```
C
C
C     ALGORITHM COMPUTE_ADJUST_BY_INDIRECT_OBSERVATIONS
C
C
C     THE CALCULATION OF OPEN TRAVERSE
C
C
      CALL CDMSR ( NUM360,NUMO,NUMO,AN360R )
      XGD = EGRID ( 2 )
      YGD = NGRID ( 2 )
      AZFIR = AZ21
      DO  240  I = 1,COUNTD
          AZIMUT = AZFIR + ANG ( I )
          IF ( AZIMUT .LT. AN360R ) GO TO 230
               AZIMUT = AZIMUT - AN360R
 230      CONTINUE
          COAZR ( I ) = AZIMUT
          CALL CRDMS ( AZIMUT,DUM1D,DUM1M,DUM1S )
          CFAZD ( I ) = IDINT ( DUM1D )
          CFAZM ( I ) = IDINT ( DUM1M )
          CFAZS ( I ) = DUM1S
          DISTAN = DIST ( I )
          CALL CXYC ( XGD,YGD,DISTAN,AZIMUT,DUMMY1,DUMMY2 )
          DELTAX ( I ) = DUMMY1 - XGD
          DELTAY ( I ) = DUMMY2 - YGD
          SUMDX = SUMDX + DELTAX ( I )
          SUMDY = SUMDY + DELTAY ( I )
          CEGRID ( I ) = DUMMY1
          CNGRID ( I ) = DUMMY2
          CALL CTANA ( DUMMY1,DUMMY2,XGD,YGD,AZC,DUM1S )
          CALL CRDMS ( AZC,DUM1D,DUM1M,DUM1S )
          CBAZD ( I ) = IDINT ( DUM1D )
          CBAZM ( I ) = IDINT ( DUM1M )
          CBAZS ( I ) = DUM1S
          AZFIR = AZC
          XGD = DUMMY1
          YGD = DUMMY2
 240  CONTINUE
C
C
C     PRINT DETAIL OF PRECALCULATED OF TRAVERSE STATION POSITIONS
C
C
      IF ( OPT .EQ. 1 ) GO TO 250
           WRITE ( 6,1180 )
           WRITE ( 6,1190 )
      GO TO 260
```

```
250   CONTINUE
            WRITE ( 6,1170 )
            WRITE ( 6,1200 )
260   CONTINUE
      WRITE ( 6,1030 )
      WRITE ( 6,1040 )
      WRITE ( 6,1050 )
      WRITE ( 6,1060 )
      WRITE ( 6,1210 ) (STNAME(2,K1),K1=1,5),NGRID(2),EGRID(2)
      DO  300 I = 1,COUNTD
          K4 = I
          K5 = I + 1
          WRITE ( 6,1220 ) K4,K5,CFAZD(I),CFAZM(I),CFAZS(I)
          IF ( OPT .EQ. 1 ) GO TO 280
                IF ( I .NE. COUNTD ) GO TO 270
                      WRITE ( 6,1070 ) K5,(STNAME(3,K1),K1=1,5),CNGRID(I),
     *                CEGRID(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
270             CONTINUE
280       CONTINUE
          IF ( I .EQ. COUNTD ) GO TO 290
                WRITE ( 6,1070 ) K5,(TRNAME(I,K1),K1=1,5),CNGRID(I),CEGRI
     *          D(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
290       CONTINUE
300   CONTINUE
      IF ( OPT .NE. 1 ) GO TO 310
            WRITE ( 6,1070 ) K5,(TRNAME(COUNTD,K1),K1=1,5),CNGRID(COUNTD)
     *      ,CEGRID(COUNTD),K5,K4,CBAZD(COUNTD),CBAZM(COUNTD),CBAZS(COUNT
     *      D)
310   CONTINUE
      IF ( OPT .EQ. 1 ) GO TO 370
            AZLAS = AZFIR + ANG ( COUNT )
            IF ( AZLAS .LT. AN360R ) GO TO 320
                  AZLAS = AZLAS - AN360R
320         CONTINUE
            K4 = I
            K5 = I + 1
            CALL CRDMS ( AZLAS,DUM1D,DUM1M,DUM1S )
            TEMD = IDINT ( DUM1D )
            TEMM = IDINT ( DUM1M )
            WRITE ( 6,1220 ) K4,K5,TEMD,TEMM,DUM1S
            WRITE ( 6,1230 ) K5,(STNAME(4,K1),K1=1,5)
            DIFAZI = AZLAS - AZ34
            WPRED ( 1 ) = DIFAZI
            CORAZ = DIFAZI / DFLOAT ( COUNTA )
            CALL CRDMS ( DIFAZI,DUM1D,DUM1M,DUM1S )
            EAZ1D = IDINT ( DABS ( DUM1D ))
            EAZ1M = IDINT ( DABS ( DUM1M ))
            EAZ1S = DABS ( DUM1S )
```

```
      CALL CRDMS ( CORAZ,DUM1D,DUM1M,DUM1S )
      EAZ2D = IDINT ( DABS ( DUM1D ))
      EAZ2M = IDINT ( DABS ( DUM1M ))
      EAZ2S = DABS ( DUM1S )
      WPRED ( 2 ) = SUMDX - ( EGRID(3) - EGRID(2) )
      WPRED ( 3 ) = SUMDY - ( NGRID(3) - NGRID(2) )
      DIFDIS = DSQRT ( WPRED(2)**2+WPRED(3)**2 )
      ERR = IDINT ( TODIS / DIFDIS )
      WRITE ( 6,1240 ) EAZ1D,EAZ1M,EAZ1S
      WRITE ( 6,1250 ) EAZ2D,EAZ2M,EAZ2S
      WRITE ( 6,1260 ) TODIS
      WRITE ( 6,1270 ) DIFDIS
      WRITE ( 6,1280 ) ERR
      ASP = COUNTD - 1
      DO 330 I = 1, ASP
         ASNY ( I ) = CNGRID ( I )
         ASEX ( I ) = CEGRID ( I )
330   CONTINUE
C
C
C     CALL SUBROUTINE LSTR TO ADJUSTED BY INDIRECT OBSERVATIONS METHOD
C
C
      NT = COUNTA + COUNTD
      NA = CCUNTA
      ND = COUNTD
      NP = ASP
      NP2 = ASP*2
      NI = NP2**2 + 3*NP2
      CALL LSTR (NT,NA,ND,NP,NP2,NI,AZ21,AZ34,NGRID,EGRID,ANG,STDA,
     *DIST,STDD,PRINTD,PW,WM,STDO,FM,BM,BMT,BMTWM,NM,TM,DELTA,NMI,
     *WK10,ASNY,ASEX,VM,STDAD)
      WRITE ( 6,1000 )
      WRITE ( 6,1290 )
      WRITE ( 6,1300 )
      WRITE ( 6,1660 )
      WRITE ( 6,1710 )
      WRITE ( 6,1310 )
      WRITE ( 6,1320 )
      DO 340 I = 1,ASP
         K4 = I*2
         K5 = I + ( I-1 )
         WRITE (6,1330) I,(TRNAME(I,K1),K1=1,5),ASNY(I),TM(K4),
     *   ASEX(I),TM(K5)
340   CONTINUE
      WRITE ( 6,1000 )
      WRITE ( 6,1720 ) STDAD
      WRITE ( 6,1730 )
```

```
      DO 350 I = 1,NA
         CALL CRDMS ( VM(1,I),DUM1D,DUM1M,DUM1S )
         DUMMY1 = DUM1D*3600.0D0 + DUM1M*60.0D0 + DUM1S
         WRITE ( 6,1740 ) I,DUMMY1
350   CONTINUE
      WRITE ( 6,1750 )
      DO 360 I = 1,ND
         K4 = NA + I
         WRITE ( 6,1760 ) I,VM ( 1,K4 )
360   CONTINUE
370   CONTINUE
      WRITE ( 6,1000 )
C
C
C
C     END COMPUTE_ADJUST_BY_INDIRECT_OBSERVATIONS
C
C
      STOP
C
C
1000 FORMAT ( '1' )
1010 FORMAT ( ///,23X,'KNOWN HORIZONTAL STATION POSITIONS' )
1020 FORMAT (     23X,'***********************************', // )
1030 FORMAT ( 31X,'UTM GRID COORDINATES',6X,'AZIMUTH   GRID AZIMUTH',/)
1040 FORMAT ( 28X,'GRID NORTHING  GRID EASTING  FROM TO',4X,'CLOCKWISE'
     *,/ )
1050 FORMAT ( 1X,'NO',3X,'NAME OF STATIONS',35X,'NO',3X,'NO',4X,'FROM N
     *ORTH', / )
1060 FORMAT ( 32X,'METERS',8X,'METERS',14X,'D. M.  S.',/ )
1070 FORMAT ( 1X,I2,1X,5A4,3X,F14.5,2X,F12.5,2X,I2,' ->',I2,1X,I3,1X,I2
     *,1X,F8.5,/ )
1080 FORMAT ( ///,31X,'OBSERVED DATA' )
1090 FORMAT (     31X,'*************',// )
1100 FORMAT ( 29X,'OBSERVED',19X,'  GRID  ',/ )
1110 FORMAT ( 6X,'NAME OF STATIONS',8X,'ANGLE',8X,'STD.',9X,'DISTANCE',
     *5X,'STD.',/ )
1120 FORMAT ( 26X,'D. M.  S.', 9X,'S.',11X,'METERS',7X,'M.' )
1130 FORMAT ( /,4X,5A4,/ )
1140 FORMAT ( 4X,5A4,1X,I3,1X,I2,1X,F8.5,1X,F8.5 )
1150 FORMAT ( 52X,F14.5,1X,F8.5 )
1160 FORMAT ( 4X,5A4,/ )
1170 FORMAT ( ///,17X,'CALCULATION OF OPEN TRAVERSE STATION POSITIONS')
1180 FORMAT ( ///,15X,'PRECALCULATED OF CLOSED TRAVERSE STATION POSITIO
     *NS' )
1190 FORMAT ( 15X,'**************************************************',
     */// )
1200 FORMAT ( 17X,'**********************************************',// )
```

```
1210 FORMAT ( 2X,'1',1X,5A4,3X,F14.5,2X,F12.5,/ )
1220 FORMAT ( 57X,I2,' ->',I2,1X,I3,1X,I2,1X,F8.5,/ )
1230 FORMAT ( 1X,I2,1X,5A4,/ )
1240 FORMAT ( ///,2X,'TOTAL AZIMUTH ERROR',8X,'= ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1250 FORMAT ( /,2X,'AZIMUTH ERROR PER STATIONS = ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1260 FORMAT ( /,'  TOTAL GRID DISTANCE         = ',F16.5,' METERS.' )
1270 FORMAT ( /,2X,'DISTANCE ERROR',13X,'= ',F10.5,' METERS. ' )
1280 FORMAT ( /,2X,'ACCURACY',18X,' =  1:',I12 )
1290 FORMAT ( ///,1X,'CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
    *BY LEAST-SQUARES ADJUSTMENT')
1300 FORMAT (      1X,'*********************************************
    ****************************',/  )
1310 FORMAT ( 28X,'GRID NORTHING',5X,'STD.',4X,'GRID EASTING',5X,'STD.'
    *,/ )
1320 FORMAT ( 1X,'NO',3X,'NAME OF STATIONS',10X,'METERS',9X,'M.',8X,'ME
    *TERS',9X,'M.',// )
1330 FORMAT ( 1X,I2,1X,5A4,3X,F14.5,2X,F9.5,2X,F12.5,2X,F9.5,/)
1340 FORMAT ( I1 )
1350 FORMAT ( I2 )
1360 FORMAT ( F20.12 )
1370 FORMAT ( 5A4 )
1380 FORMAT ( 10X,'**********************************************
    ********' )
1390 FORMAT ( 10X,'*',57X,'*' )
1400 FORMAT ( 10X,'*',9X,'THIS IS A PROGRAM TO CALCULATION OF TRAVERSE'
    *,4X,'*'/,10X,'*',4X,'STATION POSITIONS. THERE ARE TWO OPTIONS IN T
    *HIS',5X,'*',/,10X,'*',4X,'PROGRAM.:-',43X,'*' )
1410 FORMAT ( 10X,'*',4X,'1 CALCULATION OF OPEN TRAVERSE STATION POSITI
    *ONS.',4X,'*' )
1420 FORMAT ( 10X,'*',4X,'2 CALCULATION OF CLOSED TRAVERSE STATION POSI
    *TIONS',3X,'*' )
1430 FORMAT (   /,28X,'WHAT IS YOUR OPTION ?' )
1440 FORMAT (   ,20X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/   )
1450 FORMAT ( //,16X,'THERE ARE TWO OPTIONS TO ENTER DATA.',/ )
1460 FORMAT (      14X,'1 ENTER DATA BY DATA FILE.',/ )
1470 FORMAT (14X,'2 ENTER DATA BY INTERACTIVE FROM TERMINAL KEYBOARD.')
1480 FORMAT ( /,21X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/ )
1490 FORMAT ( //,16X,'WHAT IS THE NAME OF KNOWN POSITION # ',I1,'.?')
1500 FOPMAT ( /,12X,'THIS KNOWN POSITION WAS USED FOR THE FIRST BACKSIG
    *HT.')
1510 FORMAT ( /,12X,'THIS KNOWN POSITION WAS USED FOR THE LAST FORESIGH
    *T.')
1520 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE FIRST POSITION AT WHICH
    *',/,12X,'AN ANGLE WAS OBSERVED.' )
1530 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE LAST POSITION AT WHICH'
    *,/,12X,'AN ANGLE WAS OBSERVED.' )
```

```
1540 FORMAT ( /,12X,'FOR EXAMPLE : GRID NORTHING 4,072,555.85206 METERS
    *.',/,26X,'ENTER 4072555.85206ODO' )
1550 FORMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION OF THE OBSERVED AN
    *GLE',/,12X,'AT THIS STATION IN SECONDS ?',/,12X,'FOR EXAMPLE STD.
    *= 3.9 SECONDS. : ENTER 3.90D0 ' )
1560 FORMAT ( /,12X,'FOR EXAMPLE : GRID EASTING 608,279.04404 METERS.',
    */,26X,'ENTER 608279.04404D0' )
1570 FORMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION IN METERS.?',/,12X
    *,'FOR EXAMPLE  STD. = .011 METERS : ENTER 0.011ODO' )
1580 FORMAT ( /,12X,'HOW MANY POSITIONS DID YOU OBSERVE ANGLES ?',//,19
    *X,'PLEASE ENTER 01 TO 32' )
1590 FORMAT ( /,12X,'ENTER THE OBSERVED ANGLE AT THE FIRST KNOWN POSITI
    *ON.' )
1600 FORMAT ( /,10X,'ENTER THE OBSERVED ANGLE AT TRAVERSE STATION POSIT
    *ION # ',I2,'.' )
1610 FORMAT ( /,10X,'FOR EXAMPLE : OBSERVED ANGLE CLOCKWISE FROM BACKSI
    *GHT TO FORESIGHT',/,24X,'IS 169 D. 32 M. 11.11 S.',//,24X,'ENTER 1
    *69.32111110D0' )
1620 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN THE FIRST KNOWN POSITI
    *ON AND',/,12X,'TRAVERSE STATION POSITION # 1.' )
1630 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN TRAVERSE STATION POSIT
    *ION # ',I2,'.',/,12X,'AND TRAVERSE STATION POSITION # ',I2,'.'
    * )
1640 FORMAT ( /,12X,'FOR EXAMPLE : THE GRID DISTANCE BETWEEN THE STATIO
    *NS IS',/,26X,'399.052 METERS.',//,26X,'ENTER 399.052ODO' )
1650 FORMAT ( //,12X,'WHAT IS THE NAME OF TRAVERSE STATION POSITION # '
    *,I2,'.' )
1660 FORMAT ( 23X,'BY INDIRECT OBSERVATIONS METHOD',// )
1670 FORMAT ( //,12X,'WHAT IS UTM GRID NORTHING AT THIS POSITION ? ' )
1680 FORMAT ( //,12X,'WHAT IS UTM GRID EASTING AT THIS POSITION ? ' )
1690 FORMAT ( //,16X,'WHAT IS UTM GRID AZIMUTH FROM',/,12X,5A4,' TO ',
    *5A4 )
1700 FORMAT ( //,12X,'FOR EXAMPLE : UTM GRID AZIMUTH FROM  A.  TO  B.',
    */,26X,'IS 139 D. 19 M. 29.3 S. : ENTER 139.19293OD0 ' )
1710 FORMAT ( 42X,'UTM GRID COORDINATES',/)
1720 FORMAT ( //,1X,'THE STANDARD DEVIATION OF UNIT WEIGHT IS  ',F15.10
    * )
1730 FORMAT ( //,1X,'THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE ',/
    * )
1740 FORMAT ( 1X,'ANGLE # ',I2,' = ',F17.10,'  SECONDS',/ )
1750 FORMAT ( /,1X,'THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE'
    *,/)
1760 FORMAT ( 1X,'DISTANCE # ',I2,' = ',F16.10,' METERS',/ )
C
C
      END
C
C
```

```
$EJECT
C                 LS WATFIV A1
C
C
      SUBROUTINE LSTR ( NT,NA,ND,NP,NP2,NI,AZ21,AZ34,NGRID,EGRID,ANG,STD
     *A,DIST,STDD,PRINTD,PW,WM,STDO,FM,BM,BMT,BMTWM,NM,TM,DELTA,NMI,WK10
     *,ASNG,ASEG,VM,STAND)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN.                               C
C        DATE : AUGUST 11, 1984                                        C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      P
C       THIS IS A SUBROUTINE TO ADJUSTMENT OF CLOSED TRAVERSE BY        P
C                    INDIRECT OBSERVATIONS METHOD                      P
C                                                                      P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                      V
C                 VARIABLE DEFINITIONS                                 V
C                                                                      V
C     NT         = THE TOTAL NUMBER OF OBSERVATION EQUATIONS            V
C     NA         = THE TOTAL NUMBER OF OBSERVED ANGLES                  V
C     ND         = THE TOTAL NUMBER OF MEASURED DISTANCES               V
C     NP         = THE TOTAL NUMBER OF INTERMEDIATE TRAVERSE STATION    V
C                  POSITION                                            V
C     AZ21       = FIXED INITIAL AZIMUTH                               V
C     AZ34       = FIXED CLOSED AZIMUTH                                V
C     NGRID(4), EGRID(4) = FIXED COORDINATES                           V
C     ANG(NA)    = OBSERVED ANGLES                                     V
C     STDA(NA) = STANDARD DEVIATION OF OBSERVED ANGLES                 V
C     DIST(ND) = MEASURED DISTANCES                                    V
C     STDD(ND) = STANDARD DEVIATION OF MEASURED DISTANCES              V
C     FM(NT)     = B MATRIX                                            V
C     WM(NT,NT)= WEIGHT MATRIX                                         V
C     BM(NT,NP2) = A MATRIX                                            V
C     DELTA(NP2)= X MATRIX                                             V
C     TRFM       = A SUBROUTINE TO COMPUTE B MATRIX                    V
C     VMULFF     = A SUBROUTINE TO COMPUTE THE MULTIPLICATION OF MATRIX V
C     LINV2F     = A SUBROUTINE FOR INVERSION OF MATRIX                V
C     USWFM      = A SUBROUTINE FOR PRINT THE OUTPUT IN MATRIX FORM    V
C                                                                      V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                      C
C                   VARIABLE DECLARATIONS                              C
C                                                                      C
      DOUBLE PRECISION DUMMY1,NGRID(4),EGRID(4),DUMID,DUMIM,DUMIS,ANG(NA
```

```
     *),DIST(ND),WM(NT,NT),STDO(NT),ASNG(NP),ASEG(NP),FM(NT),BM(NT,NP2),
     *TEST1,TEST2,BMT(NP2,NT),BMTWM(NP2,NT),NM(NP2,NP2),TM(NP2),DELTA(NP
     *2),NMI(NP2,NP2),WK10(NI),STDA(NA),STDD(ND),DUMMY2,NUMO,AZ21,AZ34,
     *VM(1,NT),STAND
C
      INTEGER I,J,K1,IER,PRINTD,PW,CHECK
C
      DATA K1/0/,TEST1/0.00000100000D0/,NUMO/0.0D0/
C
      NC4 = 4
      NP5 = 6
      N1 = 1
C
C     SET WEIGHT MATRIX
C
      DO 30  I = 1,NT
         DO 20  J = 1,NT
            WM ( I,J ) = 0.0D0
            IF ( I .NE. J ) GO TO 10
                 WM ( I,J ) = 1.0D0
 10         CONTINUE
 20      CONTINUE
 30   CONTINUE
      IF ( PW .EQ. 1 ) GO TO 90
C
C     STANDARD DEVIATION OF THE OBSERVED ANGLES
C
           DO 40  I = 1,NA
              DUMMY1 = STDA ( I )
              CALL CDMSR ( NUMO,NUMO,DUMMY1,DUMMY2 )
              STDO ( I ) = 1.0D0 / (( DSIN ( DUMMY2 ))**2 )
 40        CONTINUE
C
C
C     STANDARD DEVIATION OF THE OBSERVED DISTANCE
C
C
           DO 50  I = 1,ND
              J = NA + I
              DUMMY1 = STDD ( I )
              STDO ( J ) = 1.0D0 / ( DUMMY1**2 )
 50        CONTINUE
           DO 80  I = 1,NT
              DO 70  J = 1,NT
                 IF ( I .NE. J ) GO TO 60
                       WM ( I,J ) = STDO ( I )
 60              CONTINUE
 70           CONTINUE
```

```
 80        CONTINUE
 90     CONTINUE
C
C       PRINT THE RESULT OF WEIGHT MATRIX
C
        IF ( PRINTD .EQ. 0 ) GO TO 100
           WRITE ( 6,1000 )
           WRITE ( 6,1010 )
           CALL USWFM ( 'R-C.',NC4,WM,NT,NT,NT,NP5 )
           WRITE ( 6,1000 )
 100    CONTINUE
C
C
 110    CONTINUE
           K1 = K1 + 1
           CHECK = 0
C
C
C       CALL SUBROUTINE TRFM FOR COMPUTATION B MATRIX
C
           CALL TRFM ( AZ21,AZ34,NA,ND,NT,NGRID,EGRID,ANG,DIST,NP,ASNG,
      *    ASEG,FM )
C
C       CALL SUBROUTINE COEFAM FOR COMPUTATION A MATRIX
C
           CALL COEFAM ( NT,NA,ND,NP,NP2,NGRID,EGRID,ASNG,ASEG,BM )
C
C
C       THE CALCULATION OF THE TRANSPOSE OF A MATRIX
C
           DO 130 I = 1,NT
              DO 120 J = 1,NP2
                 BMT ( J,I ) = BM ( I,J )
 120          CONTINUE
 130       CONTINUE
C
C
C       THE COMPUTATION OF A MATRIX TRANSPOSE * WEIGHT MATRIX
C
           CALL VMULFF ( BMT,WM,NP2,NT,NT,NP2,NT,BMTWM,NP2,IER )
C
C
C       THE COMPUTATION OF A MATRIX TRANSPOSE * WEIGHT MATRIX * A MATRIX
C
           CALL VMULFF ( BMTWM,BM,NP2,NT,NP2,NP2,NT,NM,NP2,IER )
C
C
C       THE COMPUTATION OF A MATRIX TRANSPOSE * WEIGHT MATRIX * B MATRIX
```

```
C
            CALL VMULFF ( BMTWM,FM,NP2,NT,N1,NP2,NT,TM,NP2,IER )
C
C
C     THE CALCULATION OF DELTA MATRIX ( THE CORRECTION VECTOR )
C
            CALL LINV2F ( NM,NP2,NP2,NMI,N1,WK10,IER )
            CALL VMULFF ( NMI,TM,NP2,NP2,N1,NP2,NP2,DELTA,NP2,IER )
C
C
C     UP-DATE THE APPROXIMATE VALUES
C
            DO 140 I = 1,NP
               ASEG ( I ) = ASEG ( I ) + DELTA ( ( I-1 ) * 2 + 1 )
               ASNG ( I ) = ASNG ( I ) + DELTA ( ( I-1 ) * 2 + 2 )
  140       CONTINUE
            IF ( PRINTD .EQ. 0 ) GO TO 160
                WRITE ( 6,1020 ) K1
                WRITE ( 6,1030 )
                CALL USWFM ( 'R-C.',NC4,FM,NT,NT,N1,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1040 )
                CALL USWFM ( 'R-C.',NC4,BM,NT,NT,NP2,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1050 )
                CALL USWFM ( 'R-C.',NC4,BMT,NP2,NP2,NT,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1060 )
                CALL USWFM ( 'R-C.',NC4,BMTWM,NP2,NP2,NT,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1070 )
                CALL USWFM ( 'R-C.',NC4,NM,NP2,NP2,NP2,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1080 )
                CALL USWFM ( 'R-C.',NC4,TM,NP2,NP2,N1,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1090 )
                CALL USWFM ( 'R-C.',NC4,NMI,NP2,NP2,NP2,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1100 )
                CALL USWFM ( 'R-C.',NC4,DELTA,NP2,NP2,N1,NP5 )
                WRITE ( 6,1000 )
                WRITE ( 6,1110 )
                DO 150 I = 1,NP
                   PRINT , I , '  N = ' , ASNG(I) , '  E = ' , ASEG(I)
                   PRINT , ' '
  150           CONTINUE
  160       CONTINUE
```

```
C
C
C     CHECK DELTA MATRIX
C
C
          DO 180 I = 1,NUMD
             TEST2 = DABS ( DELTA ( I ) )
             IF ( TEST2 .LT. TEST1 ) GO TO 170
                 CHECK = 1
 170         CONTINUE
 180      CONTINUE
      IF ( CHECK .NE. 0 ) GO TO 110
C
C     THE CALCULATION OF A*X MATRIX
C
      CALL VMULFF ( BM,DELTA,NT,NP2,N1,NT,NP2,STDO,NT,IER )
C
C     THE CALCULATION OF V MATRIX
C
      DO 190 I = 1,NT
         STDO ( I ) = STDO ( I ) - FM ( I )
         VM ( 1,I ) = STDO ( I )
 190  CONTINUE
      IF ( PRINTD .EQ. 0 ) GO TO 200
          WRITE ( 6,1000 )
          WRITE ( 6,1120 )
          CALL USWFM ( 'R-C.',NC4,STDO,NT,NT,N1,NP5 )
          WRITE ( 6,1000 )
          WRITE ( 6,1130 )
          CALL USWFM ( 'R-C.',NC4,VM,N1,N1,NT,NP5 )
 200  CONTINUE
C
C     THE CALCULATION OF V.T*P*V MATRIX
C
      CALL VMULFF ( WM,STDO,NT,NT,N1,NT,NT,FM,NT,IER )
      CALL VMULFF ( VM,FM,N1,NT,N1,N1,NT,STAND,N1,IER )
      IF ( PRINTD .EQ. 0 ) GO TO 210
          WRITE ( 6,1000 )
          WRITE ( 6,1140 )
          CALL USWFM ( 'R-C.',NC4,FM,NT,NT,N1,NP5 )
          WRITE ( 6,1150 )
          WRITE ( 6,1160 ) STAND
 210  CONTINUE
      CALL VMULFF ( NMI,BMT,NP2,NP2,NT,NP2,NP2,BMTWM,NP2,IER )
      STAND = DSQRT ( STAND / ( DFLOAT ( NT - NP2 )))
      DO 240 I = 1,NP2
         DO 230 J = 1,NP2
            NMI ( I,J ) = STAND * ( DSQRT ( DABS ( NMI ( I,J ))))
```

```
            IF ( I .NE. J ) GO TO 220
                  TM ( I ) = NMI ( I,J )
 220        CONTINUE
 230     CONTINUE
 240  CONTINUE
C
C     THE CALCULATION OF A*(INVERSE A.T*P*A )*A.T MATRIX
C
      CALL VMULFF ( BM,BMTWM,NT,NP2,NT,NT,NP2,WM,NT,IER )
      IF ( PRINTD .EQ. 0 ) GO TO 250
            WRITE ( 6,1000 )
            WRITE ( 6,1170 )
            CALL USWFM ( 'R-C.',NC4,WM,NT,NT,NT,NP5 )
 250  CONTINUE
      DO 280 I = 1,NT
         DO 270 J = 1,NT
            WM ( I,J ) = STAND * ( DSQRT ( DABS ( WM ( I,J ))))
            IF ( I .NE. J ) GO TO 260
                  VM ( I,I ) = WM ( I,J )
 260        CONTINUE
 270     CONTINUE
 280  CONTINUE
C
      RETURN
C
 1000 FORMAT ( '1' )
 1010 FORMAT ( ///,5X,'WEIGHT MATRIX',/ )
 1020 FORMAT ( ///,5X,I2,'  ITERATED',/ )
 1030 FORMAT ( ///,5X,'B MATRIX',/ )
 1040 FORMAT ( ///,5X,'A MATRIX',/ )
 1050 FORMAT ( ///,5X,'TRANSPOSE OF A MATRIX',/ )
 1060 FORMAT ( ///,5X,'A.T*P MATRIX',/ )
 1070 FORMAT ( ///,5X,'A.T*P*A MATRIX',/ )
 1080 FORMAT ( ///,5X,'A.T*P*B MATRIX',/ )
 1090 FORMAT ( ///,5X,'THE INVERSION OF A.T*P*A MATRIX',/ )
 1100 FORMAT ( ///,5X,'X MATRIX OR DELTA MATRIX',/ )
 1110 FORMAT ( ///,5X,'UP-DATE THE APPROXIMATE VALUES',/ )
 1120 FORMAT ( ///,5X,'V MATRIX',/ )
 1130 FORMAT ( ///,5X,'V.T MATRIX',/ )
 1140 FORMAT ( ///,5X,'P*V MATRIX',/ )
 1150 FORMAT ( ///,5X,'V.T*P*V MATRIX',/ )
 1160 FORMAT (     5X,F20.15,/ )
 1170 FORMAT ( ///,5X,'A*(INVERSE A.T*P*A)*A.T MATRIX',/)
C
C
      END
C
C
```

```
$EJECT
C                  FM WATFIV A1
C
C
      SUBROUTINE TRFM ( AZ21,AZ34,NNA,NND,NNT,YGR,XGR,OANG,ODIS,NOS,ASY
     * ,ASX,F )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     AUTHOR : LT. SAMAN AUMCHANTR RTN.                                C
C       DATE : AUGUST 11, 1984                                         C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      P
C      THIS IS A SUBROUTINE TO CALCULATION OF B MATRIX                 P
C                                                                      P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                      V
C               VARIABLE DEFINITIONS                                   V
C                                                                      V
C     AZ21    = FIXED INITIAL AZIMUTH                                  V
C     AZ34    = FIXED CLOSED AZIMUTH                                   V
C     AZF     = FORWARD AZIMUTH                                        V
C     AZB     = BACK AZIMUTH                                           V
C     NNA     = THE NUMBER OF OBSERVED ANGLES                          V
C     NND     = THE NUMBER OF MEASURED DISTANCES                       V
C     NNT     = TOTAL NUMBER OF OBSERVATION EQUATIONS                  V
C     NOS     = THE NUMBER OF THE APPROXIMATE COORDINATES              V
C     F(NNT)  = B MATRIX                                               V
C     XGR(4), YGR(4) = THE FIXED COORDINATES                           V
C     ASY(NOS), ASX(NOS) = THE APPROXIMATE COORDINATES                 V
C     TEMP1, TEMP2, TEMP3, TEMP4 = DUMMY VARIABLES                     V
C                                                                      V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                      C
C        VARIABLE DECLARATIONS                                         C
C                                                                      C
      DOUBLE PRECISION YGR(4),XGR(4),OANG(NNA),ODIS(NND),N360,A360R,DUM1
     *,NUMO,CDIS(100),TEMP1,TEMP2,TEMP3,TEMP4,ASY(NOS),AZF,AZB,ASX(NOS),
     *F(NNT),AZ21,AZ34
C
C
      INTEGER I1,I2
C
C
      DATA N360/360.0D0/,NUMO/0.0D0/
C
C
```

```
      CALL CDMSR ( N360,NUMO,NUMO,A360R )
      DO 120 I1 = 1,NNA
         IF ( I1 .NE. 1 ) GO TO 20
            TEMP1 = XGR ( 2 )
            TEMP2 = YGR ( 2 )
            AZB = AZ21
            TEMP3 = ASX ( I1 )
            TEMP4 = ASY ( I1 )
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1 )
            CDIS ( I1 ) = DUM1
            IF ( AZF .LT. AZB ) GO TO 10
               F ( I1 ) = OANG ( I1 ) - ( AZF - AZB )
            GO TO 110
10          CONTINUE
               F ( I1 ) = OANG ( I1 ) - ( A360R+AZF-AZB )
            GO TO 110
20       CONTINUE
         IF ( I1 .NE. 2 ) GO TO 50
            TEMP1 = ASX ( I1-1 )
            TEMP2 = ASY ( I1-1 )
            TEMP3 = XGR ( 2 )
            TEMP4 = YGR ( 2 )
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1 )
            TEMP3 = XGR ( 3 )
            TEMP4 = YGR ( 3 )
            IF ( NNA .EQ. 3 ) GO TO 30
               TEMP3 = ASX ( I1 )
               TEMP4 = ASY ( I1 )
30          CONTINUE
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1 )
            CDIS ( I1 ) = DUM1
            IF ( AZF .LT. AZB ) GO TO 40
               F ( I1 ) = OANG ( I1 ) - ( AZF - AZB )
            GO TO 110
40          CONTINUE
               F ( I1 ) = OANG ( I1 ) - ( A360R+AZF-AZB )
            GO TO 110
50       CONTINUE
         IF ( I1 .NE. NNA ) GO TO 70
            TEMP1 = XGR ( 3 )
            TEMP2 = YGR ( 3 )
            AZF = AZ34
            TEMP3 = ASX ( NOS )
            TEMP4 = ASY ( NOS )
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1 )
            IF ( AZF .LT. AZB ) GO TO 60
               F ( I1 ) = OANG ( I1 ) - ( AZF - AZB )
            GO TO 110
```

```
60          CONTINUE
                 F ( I1 ) = OANG ( I1 ) - ( A360R+AZF-AZB )
            GO TO 110
70      CONTINUE
            I2 = NNA - 1
            IF ( I1 .NE. I2 ) GO TO 90
                TEMP1 = ASX ( I2-1 )
                TEMP2 = ASY ( I2-1 )
                TEMP3 = ASX ( I2-2 )
                TEMP4 = ASY ( I2-2 )
                CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1 )
                TEMP3 = XGR ( 3 )
                TEMP4 = YGR ( 3 )
                CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1 )
                CDIS ( I1 ) = DUM1
                IF ( AZF .LT. AZB ) GO TO 80
                    F ( I1 ) = OANG ( I1 ) - ( AZF-AZB )
                GO TO 110
80              CONTINUE
                    F ( I1 ) = OANG ( I1 ) - ( A360R+AZF-AZB )
            GO TO 110
90          CONTINUE
            TEMP1 = ASX ( I1-1 )
            TEMP2 = ASY ( I1-1 )
            TEMP3 = ASX ( I1-2 )
            TEMP4 = ASY ( I1-2 )
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZB,DUM1 )
            TEMP3 = ASX ( I1 )
            TEMP4 = ASY ( I1 )
            CALL CTANA ( TEMP1,TEMP2,TEMP3,TEMP4,AZF,DUM1 )
            CDIS ( I1 ) = DUM1
            IF ( AZF .LT. AZB ) GO TO 100
                F ( I1 ) = OANG ( I1 ) - ( AZF-AZB )
            GO TO 110
100         CONTINUE
                F ( I1 ) = OANG ( I1 ) - ( A360R+AZF-AZB )
110     CONTINUE
120   CONTINUE
      I1 = NNA
      DO 130 I2 = 1,NND
         I1 = I1 + 1
         F ( I1 ) = ODIS ( I2 ) - CDIS ( I2 )
130   CONTINUE
C
C
      RETURN
C
      END
```

```
C
$EJECT
C                   CAM WATFIV A1
C
C
      SUBROUTINE COEFAM ( NT,NA,ND,NP,NP2,KNY,KNX,ASY,ASX,AM )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C      AUTHOR : LT. SAMAN AUMCHANTR RTN                                 C
C        DATE : AUGUST 21, 1984                                         C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       P
C       THIS IS A SUBROUTINE FOR CALCULATION OF A (COEFFICIENT) MATRIX  P
C                                                                       P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                       V
C                 VARIABLE DEFINITIONS                                  V
C                                                                       V
C     NA       = THE NUMBER OF OBSERVED ANGLES                          V
C     ND       = THE NUMBER OF MEASURED DISTANCES                       V
C     NT       = TOTAL NUMBER OF OBSERVATION EQUATIONS                  V
C     NP       = THE NUMBER OF THE APPROXIMATE COORDINATES              V
C     KNY(4),KNX(4) = THE KNOWN POSITIONS                               V
C     ASY(NP),ASX(NP) = THE APPROXIMATE VALUES OF UNKNOWN COORDINATES   V
C     AM(NT,NP2) = A MATRIX                                             V
C     DUMMY1,DUMMY2,DUMMY3,DUMMY4,DUMMY5,DUMMY6 = DUMMY VARIABLES       V
C                                                                       V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                       C
C       VARIABLE DECLARATIONS                                           C
C                                                                       C
      DOUBLE PRECISION KNY(4),KNX(4),ASY(NP),ASX(NP),AM(NT,NP2),DUMMY1,
     *DUMMY2,DUMMY3,DUMMY4,DUMMY5,DUMMY6
C
      INTEGER I1,I2,I3,I4,I5,I6,I7,J1,J2,K1,NP1
C
      NP1 = NP2 - 1
      K1 = NA - 1
      DO 160 I1 = 1, NA
         I3 = I1 - 1
         I4 = I1 - 2
         I5 = I4 + ( I4 - 1 )
         I6 = I4 + ( I4 + 1 )
         I7 = I4 + ( I4 + 3 )
         DO 150 J1 = 1, NP1, 2
            J2 = J1 + 1
```

```
      AM ( I1,J1 ) = 0.0D0
      AM ( I1,J2 ) = 0.0D0
      IF ( I1 .NE. 1 ) GO TO 20
         IF ( J1 .NE. 1 ) GO TO 10
            DUMMY1 = ASY ( 1 ) - KNY ( 2 )
            DUMMY2 = ASX ( 1 ) - KNX ( 2 )
            DUMMY3 = DUMMY1**2 + DUMMY2**2
            AM ( I1,J1 ) = ( DUMMY1 / DUMMY3 )
            AM ( I1,J2 ) = - ( DUMMY2 / DUMMY3 )
10       CONTINUE
      GO TO 150
20    CONTINUE
      IF ( I1 .NE. NA ) GO TO 40
         IF ( J1 .NE. NP1 ) GO TO 30
            DUMMY1 = ASY ( NP ) - KNY ( 3 )
            DUMMY2 = ASX ( NP ) - KNX ( 3 )
            DUMMY3 = DUMMY1**2 + DUMMY2**2
            AM ( I1,J1 ) = - ( DUMMY1 / DUMMY3 )
            AM ( I1,J2 ) = ( DUMMY2 / DUMMY3 )
30       CONTINUE
      GO TO 150
40    CONTINUE
      IF ( I1 .NE. 2 ) GO TO 80
         IF ( J1 .NE. 1 ) GO TO 60
            DUMMY1 = KNY ( 3 ) - ASY ( I3 )
            DUMMY2 = KNX ( 3 ) - ASX ( I3 )
            IF ( NA .EQ. 3 ) GO TO 50
               DUMMY1 = ASY ( I1 ) - ASY ( I3 )
               DUMMY2 = ASX ( I1 ) - ASX ( I3 )
50          CONTINUE
            DUMMY3 = DUMMY1**2 + DUMMY2**2
            DUMMY4 = KNY ( 2 ) - ASY ( I3 )
            DUMMY5 = KNX ( 2 ) - ASX ( I3 )
            DUMMY6 = DUMMY4**2 + DUMMY5**2
            AM ( I1,J1 ) = -(DUMMY1/DUMMY3)+(DUMMY4/DUMMY6)
            AM ( I1,J2 ) = (DUMMY2/DUMMY3)-(DUMMY5/DUMMY6)
         GO TO 150
60       CONTINUE
         IF ( J1 .NE. 3 ) GO TO 70
            AM ( I1,J1 ) = ( DUMMY1 / DUMMY3 )
            AM ( I1,J2 ) = - ( DUMMY2 / DUMMY3 )
70       CONTINUE
      GO TO 150
80    CONTINUE
      IF ( I1 .NE. K1 ) GO TO 110
         IF ( J1 .NE. I5 ) GO TO 90
            DUMMY1 = ASY ( I4 ) - ASY ( I3 )
            DUMMY2 = ASX ( I4 ) - ASX ( I3 )
```

```
                 DUMMY3 = DUMMY1**2 + DUMMY2**2
                 AM ( I1,J1 ) = - ( DUMMY1 / DUMMY3 )
                 AM ( I1,J2 ) = ( DUMMY2 / DUMMY3 )
             GO TO 150
90           CONTINUE
             IF ( J1 .NE. I6 ) GO TO 100
                 DUMMY1 = KNY ( 3 ) - ASY ( I3 )
                 DUMMY2 = KNX ( 3 ) - ASX ( I3 )
                 DUMMY3 = DUMMY1**2 + DUMMY2**2
                 DUMMY4 = ASY ( I4 ) - ASY ( I3 )
                 DUMMY5 = ASX ( I4 ) - ASX ( I3 )
                 DUMMY6 = DUMMY4**2 + DUMMY5**2
                 AM ( I1,J1 ) = -(DUMMY1/DUMMY3)+(DUMMY4/DUMMY6)
                 AM ( I1,J2 ) = (DUMMY2/DUMMY3)-(DUMMY5/DUMMY6)
100          CONTINUE
         GO TO 150
110      CONTINUE
         IF ( J1 .NE. I5 ) GO TO 120
             DUMMY1 = ASY ( I4 ) - ASY ( I3 )
             DUMMY2 = ASX ( I4 ) - ASX ( I3 )
             CUMMY3 = DUMMY1**2 + DUMMY2**2
             AM ( I1,J1 ) = - ( DUMMY1 / DUMMY3 )
             AM ( I1,J2 ) = ( DUMMY2 / DUMMY3 )
         GO TO 150
120      CONTINUE
         IF ( J1 .NE. I6 ) GO TO 130
             CUMMY1 = ASY ( I1 ) - ASY ( I3 )
             DUMMY2 = ASX ( I1 ) - ASX ( I3 )
             CUMMY3 = DUMMY1**2 + DUMMY2**2
             DUMMY4 = ASY ( I4 ) - ASY ( I3 )
             DUMMY5 = ASX ( I4 ) - ASX ( I3 )
             DUMMY6 = DUMMY4**2 + DUMMY5**2
             AM ( I1,J1 ) = -(DUMMY1/DUMMY3)+(DUMMY4/DUMMY6)
             AM ( I1,J2 ) = (DUMMY2/DUMMY3)-(DUMMY5/DUMMY6)
         GO TO 150
130      CONTINUE
         IF ( J1 .NE. I7 ) GO TO 140
             DUMMY1 = ASY ( I1 ) - ASY ( I3 )
             DUMMY2 = ASX ( I1 ) - ASX ( I3 )
             CUMMY3 = DUMMY1**2 + DUMMY2**2
             AM ( I1,J1 ) = ( DUMMY1 / DUMMY3 )
             AM ( I1,J2 ) = - ( DUMMY2 / DUMMY3 )
140      CONTINUE
150    CONTINUE
160 CONTINUE
    DO 240 I1 = 1, ND
       I2 = NA + I1
       I3 = I1 - 1
```

```
            I4 = I3 + ( I3 - 1 )
            I5 = I3 + ( I3 + 1 )
            DO 230 J1 = 1, NP1, 2
               J2 = J1 + 1
               AM ( I2,J1 ) = 0.0D0
               AM ( I2,J2 ) = 0.0D0
               IF ( I1 .NE. 1 ) GO TO 180
                  IF ( J1 .NE. 1 ) GO TO 170
                     DUMMY1 = KNY ( 2 ) - ASY ( 1 )
                     DUMMY2 = KNX ( 2 ) - ASX ( 1 )
                     DUMMY3 = DSQRT ( DUMMY1**2 + DUMMY2**2 )
                     AM ( I2,J1 ) = - ( DUMMY2 / DUMMY3 )
                     AM ( I2,J2 ) = - ( DUMMY1 / DUMMY3 )
  170             CONTINUE
               GO TO 230
  180          CONTINUE
               IF ( I1 .NE. ND ) GO TO 200
                  IF ( J1 .NE. NP1 ) GO TO 190
                     DUMMY1 = ASY ( NP ) - KNY ( 3 )
                     DUMMY2 = ASX ( NP ) - KNX ( 3 )
                     DUMMY3 = DSQRT ( DUMMY1**2 + DUMMY2**2 )
                     AM ( I2,J1 ) = ( DUMMY2 / DUMMY3 )
                     AM ( I2,J2 ) = ( DUMMY1 / DUMMY3 )
  190             CONTINUE
               GO TO 230
  200          CONTINUE
               IF ( I4 .NE. J1 ) GO TO 210
                  DUMMY1 = ASY ( I3 ) - ASY ( I1 )
                  DUMMY2 = ASX ( I3 ) - ASX ( I1 )
                  DUMMY3 = DSQRT ( DUMMY1**2 + DUMMY2**2 )
                  AM ( I2,J1 ) = ( DUMMY2 / DUMMY3 )
                  AM ( I2,J2 ) = ( DUMMY1 / DUMMY3 )
               GO TO 230
  210          CONTINUE
               IF ( I5 .NE. J1 ) GO TO 220
                  DUMMY1 = ASY ( I3 ) - ASY ( I1 )
                  DUMMY2 = ASX ( I3 ) - ASX ( I1 )
                  DUMMY3 = DSQRT ( DUMMY1**2 + DUMMY2**2 )
                  AM ( I2,J1 ) = - ( DUMMY2 / DUMMY3 )
                  AM ( I2,J2 ) = - ( DUMMY1 / DUMMY3 )
  220          CONTINUE
  230       CONTINUE
  240    CONTINUE
C
         RETURN
C
         END
```

APPENDIX C

PROGRAM 3

COMPUTER OUTPLT

THE CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
BY CONDITION EQUATIONS METHOD
FOR UNCORRELATED OBSERVATIONS WITH EQUAL PRECISION

KNOWN HORIZONTAL STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	UTM GRID COORDINATES GRID EASTING METERS	AZIMUTH FROM NO	AZIMUTH TO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	PIPHER					
2	MOSS 2	4072555.85206	608279.04404	2 ->	1	100 16 23.77800
3	HOLM	4079258.31754	612238.85256	3 ->	4	136 33 26.33400
4	MORAN					

OBSERVED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	STD. S.	GRID DISTANCE METERS	STD. M.
PIPHER				
MOSS 2	246 5 43.20000	1.98400	1424.00400	0.00100
MOSSBACK	222 51 8.60000	1.40500	365.74400	0.00100
DUNE TEMP	190 15 2.60000	1.20300	6476.27100	0.00300
HOLM	277 5 17.00000	1.61400		
MORAN				

PRECALCULATED OF CLOSED TRAVERSE STATION POSITIONS

NC	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM NO TO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 6.97800
2	MOSSBACK	4073939.74473	607943.44238	2 -> 1	166 22 6.97800
				2 -> 3	29 13 15.57800
3	DUNE TEMP	4074258.94534	608121.99110	3 -> 2	209 13 15.57800
				3 -> 4	39 28 18.17800
4	HOLM	4079258.22818	612238.93865	4 -> 3	219 28 18.17800
				4 -> 5	136 33 35.17800
5	MORAN				

TOTAL AZIMUTH ERROR = 0 D. 0 M. 8.84400 S.
AZIMUTH ERROR PER STATIONS = 0 D. 0 M. 2.21100 S.
TOTAL GRID DISTANCE = 8266.01900 METERS.
DISTANCE ERROR = 0.12408 METERS.
ACCURACY = 1: 66617

CORRECTED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	GRID DISTANCE METERS
PIPHER		
MOSS 2	246 5 45.81717	1424.00400
MOSSBACK	222 51 5.36399	365.74400
DUNE TEMP	190 14 58.98196	6476.27100
HOLM	277 5 12.39287	
MORAN		

```
CALCULATION OF CLOSED TRAVERSE STATION POSITIONS BY LEAST-SQUARES ADJUSTMENT
****************************************************************************

                           BY CONDITION EQUATION METHOD

                              UTM GRID COORDINATES      AZIMUTH    GRID AZIMUTH
                           GRID NORTHING  GRID EASTING  FROM TO     CLOCKWISE

NO   NAME OF STATIONS                                   NO   NO     FROM NORTH

                               METERS        METERS                D. M.  S.

 1  MOSS 2                 4072555.85206  608279.04404
                                                        1 -> 2 346 22  9.59517
 2  MOSSBACK               4073939.74899  607943.45994  2 -> 1 166 22  9.59517
                                                        2 -> 3  29 13 14.95917
 3  DUNE TEMP              4074258.95013  608122.00770  3 -> 2 209 13 14.95917
                                                        3 -> 4  39 28 13.94113
 4  HOLM                   4079258.31754  612238.85256  4 -> 3 219 28 13.94113
                                                        4 -> 5 136 33 26.33400
 5  MORAN

 CHECK ROUND OFF ERROR
 AZIMUTH    =    0 D.  0 M.  0.00000 S.
 DISTANCE   =     0.00000 METERS.

THE STANDARD DEVIATION OF UNIT WEIGHT IS    0.0000201139

THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE

ANGLE #  1 =       0.1591810531  SECONDS
ANGLE #  2 =       2.9426247963  SECONDS
ANGLE #  3 =       2.9172247526  SECONDS
ANGLE #  4 =       0.1336934925  SECONDS

THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE

DISTANCE #  1 =     0.0000201139  METERS
DISTANCE #  2 =     0.0000201139  METERS
DISTANCE #  3 =     0.0000201139  METERS
```

THE CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
BY CONDITION EQUATIONS METHOD
FOR UNCORRELATED OBSERVATIONS WITH UNEQUAL PRECISION

KNOWN HORIZONTAL STATION POSITIONS
****

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	PIPHER				
2	MOSS 2	4072555.85206	608279.04404	2 -> 1	100 16 23.77800
3	HOLM	4079258.31754	612238.85256	3 -> 4	136 33 26.33400
4	MORAN				

OBSERVED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	STD. S.	GRID DISTANCE METERS	STD. M.
PIPHER				
MOSS 2	246 5 43.20000	1.98400		
			1424.00400	0.00100
MOSSBACK	222 51 8.60000	1.40500		
			365.74400	0.00100
DUNE TEMP	190 15 2.60000	1.20300		
			6476.27100	0.00300
HOLM	277 5 17.00000	1.61400		
MORAN				

PRECALCULATED OF CLOSED TRAVERSE STATION POSITIONS

NO	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	UTM GRID COORDINATES GRID EASTING METERS	AZIMUTH FROM NO TO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 6.97800
2	MOSSBACK	4073939.74473	607943.44238	2 -> 1	166 22 6.97800
				2 -> 3	29 13 15.57800
3	DUNE TEMP	4074258.94534	608121.99110	3 -> 2	209 13 15.57800
				3 -> 4	39 28 18.17800
4	HOLM	4079258.22818	612238.93865	4 -> 3	219 28 18.17800
				4 -> 5	136 33 35.17800
5	MORAN				

TOTAL AZIMUTH ERROR = 0 D. 0 M. 8.84400 S.
AZIMUTH ERROR PER STATIONS = 0 D. 0 M. 2.21100 S.
TOTAL GRID DISTANCE = 8266.01900 METERS.
DISTANCE ERROR = 0.12408 METERS.
ACCURACY = 1: 66617

CORRECTED DATA
****

NAME OF STATIONS	OBSERVED ANGLE D. M. S.	GRID DISTANCE METERS
PIPHER		
MOSS 2	246 5 45.12117	
		1424.00424
MOSSBACK	222 51 5.23554	
		365.74439
DUNE TEMP	190 14 59.95242	
		6476.27457
HOLM	277 5 12.24688	
MORAN		

CALCULATION OF CLOSED TRAVERSE STATION POSITIONS BY LEAST-SQUARES ADJUSTMENT
****
BY CONDITION EQUATION METHOD

NC	NAME OF STATIONS	UTM GRID COORDINATES GRID NORTHING METERS	UTM GRID COORDINATES GRID EASTING METERS	AZIMUTH FROM TO NO NO	GRID AZIMUTH CLOCKWISE FROM NORTH D. M. S.
1	MOSS 2	4072555.85206	608279.04404		
				1 -> 2	346 22 8.89917
2	MOSSBACK	4073939.74809	607943.45521	2 -> 1	166 22 8.89917
				2 -> 3	29 13 14.13470
3	DUNE TEMP	4074258.95029	608122.00189	3 -> 2	209 13 14.13470
				3 -> 4	39 28 14.08712
4	HOLM	4079258.31754	612238.85256	4 -> 3	219 28 14.08712
				4 -> 5	136 33 26.33400
5	MORAN				

CHECK ROUND OFF ERROR
AZIMUTH = 0 D. 0 M. 0.00000 S.
DISTANCE = 0.00000 METERS.

THE STANDARD DEVIATION OF UNIT WEIGHT IS 2.6967855476

THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE

ANGLE # 1 = 1.4033951177 SECONDS
ANGLE # 2 = 2.6409612857 SECONDS
ANGLE # 3 = 2.5525289050 SECONDS
ANGLE # 4 = 0.2613580638 SECONDS

THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE

DISTANCE # 1 = 0.0026855980 METERS
DISTANCE # 2 = 0.0026737313 METERS
DISTANCE # 3 = 0.0074503265 METERS

PROGRAM LISTING

```
C                 TRAVCON WATFIV A1
C
$JOB              SAM3,NOCHECK
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C       AUTHOR : LT. SAMAN AUMCHANTR RTN.                                C
C         DATE : JULY 11 , 1984                                          C
C                                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        P
C        THIS IS A PROGRAM FOR CALCULATION  OF TRAVERSE STATION          P
C    POSITIONS BY CONDITION EQUATION METHOD.                             P
C                                                                        P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                        V
C                 VARIABLE DEFINITIONS                                   V
C                                                                        V
C      OPT      = OPTION FOR CALCULATION OF OPEN OR CLOSED TRAVERSE      V
C      CHOIC    = OPTION FOR ENTER DATA BY DATA FILE OR INTERACTIVE      V
C                 FROM TERMINAL KEYBOARD.                                V
C      DUMMY1,DUMMY2,DUM1D,DUM1M,DUM1S = THE DUMMY VARIABLES             V
C      NGRID(4)= GRID NORTHING OF THE KNOWN POSITIONS.                   V
C      EGRID(4)= GRID EASTING OF THE KNOWN POSITIONS.                    V
C      STNAME(4,5) = THE NAME OF KNOWN STATIONS.                         V
C      TRNAME( 32,5) = THE NAME OF TRAVERSE STATIONS.                    V
C      DIRECD(4),DIRECM(4),DIRECS(4) = THE GRID AZIMUTH OF KNOWN         V
C                 POSITIONS.                                             V
C      STAT1(4),STAT2(4) = THE NUMBER OF KNOWN POSITIONS.                V
C      COUNT, COUNTA = THE NUMBER OF OBSERVED ANGLES.                    V
C      ANG( 32)= THE OBSERVED ANGLE IN RADIAN.                           V
C      ANGD( 32),ANGM( 32),ANGS( 32) = THE OBSERVED ANGLES IN DEGREE,    V
C                 MINUTE AND SECOND.                                     V
C      COUNTN, COUNTD = THE NUMBER OF MEASURED DISTANCE.                 V
C      DIST( 32) = THE MEASURED DISTANCE IN METERS.                      V
C      TODIS    = THE SUM OF MEASURED DISTANCE.                          V
C      CFAZD( 32),CFAZM( 32),CFAZS( 32) = THE COMPUTED FORWARD AZIMUTH   V
C      CBAZD( 32),CBAZM( 32),CBAZS( 32) = THE COMPUTED BACK AZIMUTH      V
C      CNGRID( 32) = GRID NORTHING OF TRAVERSE STATION POSITIONS.        V
C      CEGRID( 32) = GRID EASTING OF TRAVERSE STATION POSITIONS.         V
C      ERR      = THE TOTAL DISTANCE ERROR.                              V
C      DIFAZI   = THE TOTAL ANGLE ERROR IN RADIAN.                       V
C      EAZ1D, EAZ1M, EAZ1S = THE TOTAL ANGLE ERROR IN DEGREE,            V
C                 MINUTE AND SECOND.                                     V
C      EAZ2D, EAZ2M, EAZ2S = THE ANGLE ERROR PER STATION.                V
```

```
C       TDMS      = SUBROUTINE FOR CHANGE INPUT DATA FORM DDD.MMSSSSSS     V
C                   TO DEGREE , MINUTE AND SECOND.                         V
C       CDMSR     = SUBROUTINE FOR CHANGE DEGREE, MINUTE AND               V
C                   SECOND TO RADIAN.                                      V
C       CRDMS     = SUBROUTINE FOR CHANGE RADIAN TO DEGREE, MINUTE         V
C                   AND SECOND.                                            V
C       CTANA     = SUBROUTINE FOR CALCULATION OF GRID AZIMUTH AND THE     V
C                   THE DISTANCE BETWEEN POSITIONS # 1 AND #2              V
C       CXYC      = SUBROUTINE FOR COMPUTATION OF THE UTM GRID COORDINATES V
C       CLSQ      = SUBROUTINE FOR CALCULATION OF LEAST-SQUARES ADJUSTMENT V
C                                                                          V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                          C
C                                                                          C
C                 VARIABLE DECLARATIONS                                    C
C                                                                          C
C                                                                          C
        DOUBLE PRECISION DUMMY1,CFAZS(32 ),CBAZS(32 ),EAZ1S,EAZ2S,CEGRID(
     *32),CNGRID( 32),DIS21,AZ21,AZ12,DIS34,AZ34,DUM1D,AZ43,DUM1M,DUM1S,
     *DUMMY2,ANG( 32),DIST( 32),AZFIR,AZIMUT,NUM0,NUM360,AN360R,DISTAN,
     *AZC,DIFDIS,AZLAS,DIFAZI,CORAZ,TODIS,VAR( 64),PIBT( 64,3),NUM1
        DOUBLE PRECISION ANGS( 32),DIRECS(4),NGRID(4),EGRID(4),PMI( 64,64
     * ),XGD,YGD,STDA( 32),STDD( 32),DELTAX( 32),DELTAY( 32),SUMDX,SUMDY
     *,WPRED(3),CCAZR( 32),CORDX1,CORDY1,COBM1( 64),COBM2( 64),COBM3( 64
     *),CORAD( 64),BM(3,64 ),BMT( 64,3),DM(64,64),BPIM(3,64),QBPIM(3,64)
     *,STAND,VTM(1,64)
C
C
        INTEGER DES,OPT,COUNT,I,COUNTA,COUNTD,ERR,CHOIC,LOOP1,DIRECD(4),DI
     *RECM(4),STAT1(4),STAT2(4),STNAME(4,5),K1,K2,ANGD( 32),ANGM( 32),CO
     *UNTN,TRNAME( 32,5),K4,K5,COUNTP,EAZ1D,EAZ1M,EAZ2D,EAZ2M,CFAZD( 32)
     *,CBAZD( 32),CBAZM( 32),TEMD,TEMM,ADJ1,CFAZM( 32),J,NT,ITER1,PRINTD
     *,PW
C
C
C       IF PW = 1; IT MEANS UNWEIGHT MATRIX
C       IF PRINTD = 0; IT MEANS NO PRINT DETAILS OF MATRIX COMPUTATION
C
C
        DATA PW/0/,PRINTD/0/
C
C
        DATA NUM0/0.0D0/,NUM360/360.0D0/,TODIS/0.0D0/,NUM1/1.0D0/,SUMDY/0
     *.0D0/,SUMDX/0.0D0/
C
C
C       ALGORITHM INPUT_DATA
C
```

```
C     PRINT TITLE
C
C
      WRITE ( 6,1000 )
      WRITE ( 6,1380 )
      WRITE ( 6,1390 )
      WRITE ( 6,1400 )
      WRITE ( 6,1410 )
      WRITE ( 6,1420 )
      WRITE ( 6,1390 )
      WRITE ( 6,1380 )
      WRITE ( 6,1430 )
      WRITE ( 6,1440 )
C
C
C     READ OPTIONS  1 = OPEN TRAVERSE
C                   2 = CLOSED TRAVERSE
C
C
      READ ( 4,1340 ) OPT
C
C
      WRITE ( 6,1450 )
      WRITE ( 6,1460 )
      WRITE ( 6,1470 )
      WRITE ( 6,1430 )
      WPITE ( 6,1480 )
C
C
C     READ OPTIONS  1 = ENTER DATA BY DATA FILE
C                   2 = ENTER DATA BY INTERACTIVE
C
C
      READ ( 4,1340 ) CHOIC
C
C
C     GET UTM GRID NORTHING, EASTING, AZIMUTH, AND THE NAME
C                         OF KNOWN POSITIONS
C
C
      DES = 04
      IF ( CHOIC .EQ. 2 ) GO TO 10
           DES = 03
 10   CONTINUE
      LOOP1 = 2
      IF ( OPT .EQ. 1 ) GO TO 20
           LOOP1 = 4
 20   CONTINUE
```

```
      DO  70  I = 1,LOOP1
          WRITE ( 6,1490 ) I
          IF ( I .NE. 1 ) GO TO 30
               WRITE ( 6,1500 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
               GO TO 60
   30     CONTINUE
          IF ( I .NE. 2 ) GO TO 40
               WRITE ( 6,1520 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
               WRITE ( 6,1750 )
               WRITE ( 6,1540 )
               READ ( DES,1360 ) DUMMY1
               NGRID ( I ) = DUMMY1
               WRITE ( 6,1760 )
               WRITE ( 6,1560 )
               READ ( DES,1360 ) DUMMY1
               EGRID ( I ) = DUMMY1
               WRITE ( 6,1770 ) (STNAME(2,K1),K1=1,5),(STNAME(1,K1),K1=1
     *         ,5)
               WRITE ( 6,1780 )
               READ ( DES,1360 ) DUMMY1
               CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
               DIRECD ( 2 ) = IDINT ( DUM1D )
               DIRECM ( 2 ) = IDINT ( DUM1M )
               DIRECS ( 2 ) = DUM1S
               CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
               AZ21 = DUMMY1
               STAT1 ( 2 ) = 2
               STAT2 ( 2 ) = 1
               GO TO 60
   40     CONTINUE
          IF ( I .NE. 3 ) GO TO 50
               WRITE ( 6,1530 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
               WRITE ( 6,1750 )
               WRITE ( 6,1540 )
               READ ( DES,1360 ) DUMMY1
               NGRID ( I ) = DUMMY1
               WRITE ( 6,1760 )
               WRITE ( 6,1560 )
               READ ( DES,1360 ) DUMMY1
               EGRID ( I ) = DUMMY1
               GO TO 60
   50     CONTINUE
          IF ( I .NE. 4 ) GO TO 60
               WRITE ( 6,1510 )
               READ ( DES,1370 ) ( STNAME ( I,K1 ) , K1 = 1,5 )
```

```
            WRITE ( 6,1770 ) (STNAME(3,K1),K1=1,5),(STNAME(4,K1),K1=1
     *      ,5)
            WRITE ( 6,1780 )
            READ ( DES,1360 ) DUMMY1
            CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
            DIRECD ( 3 ) = IDINT ( DUM1D )
            DIRECM ( 3 ) = IDINT ( DUM1M )
            DIRECS ( 3 ) = DUM1S
            CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY1 )
            AZ34 = DUMMY1
            STAT1 ( 3 ) = 3
            STAT2 ( 3 ) = 4
   60     CONTINUE
   70 CONTINUE
C
C
C     READ THE NUMBER OF OBSERVED ANGLES
C
C
      WRITE ( 6,1580 )
      READ ( DES,1350 ) COUNT
      COUNTA = COUNT
C
C
C     READ THE OBSERVED ANGLES
C
C
      DO  100 I = 1,COUNTA
          IF ( I .GT. 1 ) GO TO 80
              WRITE ( 6,1590 )
              GO TO 90
   80     CONTINUE
          K2 = I -1
          WRITE ( 6,1600 ) K2
   90     CONTINUE
          WRITE ( 6,1610 )
          READ ( DES,1360 ) DUMMY1
          CALL TDMS ( DUMMY1,DUM1D,DUM1M,DUM1S )
          CALL CDMSR ( DUM1D,DUM1M,DUM1S,DUMMY2 )
          ANG ( I ) = DUMMY2
          ANGD ( I ) = IDINT ( DUM1D )
          ANGM ( I ) = IDINT ( DUM1M )
          ANGS ( I ) = DUM1S
          WRITE ( 6,1550 )
          READ ( DES,1360 ) DUMMY1
          STDA ( I ) = DUMMY1
  100 CONTINUE
C
```

```
C
C     READ THE GRID DISTANCE
C
C
      COUNTD = COUNT
      IF ( OPT .EQ. 1 ) GO TO 110
         COUNTD = COUNT - 1
 110  CONTINUE
      DO  140 I = 1,COUNTD
          IF ( I .GT. 1 ) GO TO 120
               WRITE ( 6,1620 )
               GO TO 130
 120      CONTINUE
          K4 = I - 1
          K5 = I
          WRITE ( 6,1630 ) K4,K5
 130      CONTINUE
          WRITE ( 6,1640 )
          READ ( DES,1360 ) DUMMY1
          DIST ( I ) = DUMMY1
          TODIS = TODIS + DUMMY1
          WRITE ( 6,1570 )
          READ ( DES,1360 ) DUMMY1
          STDD ( I ) = DUMMY1
 140  CONTINUE
      COUNTN = COUNTD
      IF ( OPT .EQ. 1 ) GO TO 150
         COUNTN = COUNTD - 1
 150  CONTINUE
      DO  160 I = 1,COUNTN
          WRITE ( 6,1650 ) I
          READ ( DES,1370 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
 160  CONTINUE
C
C
C     END INPUT_DATA
C
C     ALGORITHM PRINT_INPUT_DATA
C
C     PRINT DETAILS OF THE KNOWN STATION POSITIONS
C
C
      WRITE ( 6,1000 )
      WRITE ( 6,1010 )
      WRITE ( 6,1020 )
      WRITE ( 6,1030 )
      WRITE ( 6,1040 )
      WRITE ( 6,1050 )
```

```
      WRITE ( 6,1060 )
      DO  180 I = 1,LOOP1
          IF ( I .EQ. 2 .OR. I .EQ. 3 ) GO TO 170
                WRITE ( 6,1230 ) I , ( STNAME ( I,K1 ) , K1 = 1,5 )
                GO TO 180
170       CONTINUE
          WRITE ( 6,1070 ) I,(STNAME(I,K1),K1=1,5),NGRID(I),EGRID(I),STA
     *    T1(I),STAT2(I),DIRECD(I),DIRECM(I),DIRECS(I)
180   CONTINUE
      WRITE ( 6,1000 )
      WRITE ( 6,1080 )
      WRITE ( 6,1090 )
      WRITE ( 6,1100 )
      WRITE ( 6,1110 )
      WRITE ( 6,1120 )
      WRITE ( 6,1130 ) ( STNAME ( 1,K1 ) , K1 = 1,5 )
      WRITE ( 6,1140 ) (STNAME(2,K1),K1 = 1,5),ANGD(1),ANGM(1),ANGS(1),S
     *TDA(1)
      WRITE ( 6,1150 ) DIST ( 1 ),STDD ( 1 )
      COUNTP = COUNTD - 1
      IF ( COUNTP .NE. 0 ) GO TO 190
            WRITE ( 6,1160 ) ( TRNAME ( 1,K1 ) , K1 = 1,5 )
            GO TO 220
190   CONTINUE
      DO  200 I = 1,COUNTP
          K2 = I + 1
          WRITE ( 6,1140 ) (TRNAME(I,K1),K1=1,5),ANGD(K2),ANGM(K2),ANGS(
     *    K2),STDA(K2)
          WRITE ( 6,1150 ) DIST ( K2 ),STDD ( K2 )
200   CONTINUE
      IF ( OPT .NE. 1 ) GO TO 210
            WRITE ( 6,1160 ) ( TRNAME ( I,K1 ) , K1 = 1,5 )
            GO TO 220
210   CONTINUE
      WRITE ( 6,1140 ) (STNAME(3,K1),K1=1,5),ANGD(COUNTA),ANGM(COUNTA),
     *ANGS(COUNTA),STDA(COUNTA)
      WRITE ( 6,1130 ) ( STNAME ( 4,K1 ) , K1 = 1,5 )
220   CONTINUE
      WRITE ( 6,1000 )
C
C
C     END PRINT_INPUT_DATA
C
C     ALGORITHM COMPUTE_ADJUST_BY_CONDITION_EQUATIONS
C
C     THE CALCULATION OF OPEN TRAVERSE
C
C
```

```
      CALL CDMSR ( NUM360,NUMO,NUMO,AN360R )
      ITER1 = 2
      IF ( OPT .NE. 1 ) GO TO 230
         ITER1 = 1
230   CONTINUE
      DO 450 ADJ1 = 1,ITER1
         XGD = EGRID ( 2 )
         YGD = NGRID ( 2 )
         AZFIR = AZ21
         DO  250  I = 1,COUNTD
             AZIMUT = AZFIR + ANG ( I )
             IF ( AZIMUT .LT. AN360R ) GO TO 240
                  AZIMUT = AZIMUT - AN360R
240          CONTINUE
             COAZR ( I ) = AZIMUT
             CALL CRDMS ( AZIMUT,DUM1D,DUM1M,DUM1S )
             CFAZD ( I ) = IDINT ( DUM1D )
             CFAZM ( I ) = IDINT ( DUM1M )
             CFAZS ( I ) = DUM1S
             DISTAN = DIST ( I )
             CALL CXYC ( XGD,YGD,DISTAN,AZIMUT,DUMMY1,DUMMY2 )
             DELTAX ( I ) = DUMMY1 - XGD
             DELTAY ( I ) = DUMMY2 - YGD
             SUMDX = SUMDX + DELTAX ( I )
             SUMDY = SUMDY + DELTAY ( I )
             CEGRID ( I ) = DUMMY1
             CNGRID ( I ) = DUMMY2
             CALL CTANA ( DUMMY1,DUMMY2,XGD,YGD,AZC,DUM1S )
             CALL CRDMS ( AZC,DUM1D,DUM1M,DUM1S )
             CBAZD ( I ) = IDINT ( DUM1D )
             CBAZM ( I ) = IDINT ( DUM1M )
             CBAZS ( I ) = DUM1S
             AZFIR = AZC
             XGD = DUMMY1
             YGD = DUMMY2
250      CONTINUE
C
C
C    >>> PRINT DETAIL OF PRECALCULATED OF TRAVERSE STATION POSITIONS <<<
C
C
         IF ( OPT .EQ. 1 ) GO TO 270
             IF ( ADJ1 .NE. 1 ) GO TO 260
                  WRITE ( 6,1180 )
                  WRITE ( 6,1190 )
                  GO TO 280
260          CONTINUE
             WRITE ( 6,1000 )
```

```
              WRITE ( 6,1290 )
              WRITE ( 6,1300 )
              WRITE ( 6,1740 )
              GO TO 280
270       CONTINUE
          WRITE ( 6,1170 )
          WRITE ( 6,1200 )
280       CONTINUE
          WRITE ( 6,1030 )
          WRITE ( 6,1040 )
          WRITE ( 6,1050 )
          WRITE ( 6,1060 )
          WRITE ( 6,1210 ) (STNAME(2,K1),K1=1,5),NGRID(2),EGRID(2)
          DO  320 I = 1,COUNTD
             K4 = I
             K5 = I + 1
             WRITE ( 6,1220 ) K4,K5,CFAZD(I),CFAZM(I),CFAZS(I)
             IF ( OPT .EQ. 1 ) GO TO 300
                IF ( I .NE. COUNTD ) GO TO 290
                   WRITE ( 6,1070 ) K5,(STNAME(3,K1),K1=1,5),CNGRI
     *             D(I),CEGRID(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
290             CONTINUE
300          CONTINUE
             IF ( I .EQ. COUNTD ) GO TO 310
                WRITE ( 6,1070 ) K5,(TRNAME(I,K1),K1=1,5),CNGRID(I),C
     *          EGRID(I),K5,K4,CBAZD(I),CBAZM(I),CBAZS(I)
310          CONTINUE
320       CONTINUE
          IF ( OPT .NE. 1 ) GO TO 330
              WRITE ( 6,1070 ) K5,(TRNAME(COUNTD,K1),K1=1,5),CNGRID(COU
     *        NTD),CEGRID(COUNTD),K5,K4,CBAZD(COUNTD),CBAZM(COUNTD),CBA
     *        ZS(COUNTD)
330       CONTINUE
          IF ( OPT .EQ. 1 ) GO TO 440
              AZLAS = AZFIR + ANG ( COUNT )
              IF ( AZLAS .LT. AN360R ) GO TO 340
                  AZLAS = AZLAS - AN360R
340           CONTINUE
              K4 = I
              K5 = I + 1
              CALL CPDMS ( AZLAS,DUM1D,DUM1M,DUM1S )
              TEMD = IDINT ( DUM1D )
              TEMM = IDINT ( DUM1M )
              WRITE ( 6,1220 ) K4,K5,TEMD,TEMM,DUM1S
              WRITE ( 6,1230 ) K5,(STNAME(4,K1),K1=1,5)
              DIFAZI = AZLAS - AZ34
              WPRED ( 1 ) = DIFAZI
              CORAZ = DIFAZI / DFLOAT ( COUNTA )
```

```
      CALL CRDMS ( DIFAZI,DUM1D,DUM1M,DUM1S )
      EAZ1D = IDINT ( DABS ( DUM1D ))
      EAZ1M = IDINT ( DABS ( DUM1M ))
      EAZ1S = DABS ( DUM1S )
      CALL CRDMS ( CORAZ,DUM1D,DUM1M,DUM1S )
      EAZ2D = IDINT ( DABS ( DUM1D ))
      EAZ2M = IDINT ( DABS ( DUM1M ))
      EAZ2S = DABS ( DUM1S )
      WPRED ( 2 ) = SUMDX - ( EGRID(3) - EGRID(2) )
      WPRED ( 3 ) = SUMDY - ( NGRID(3) - NGRID(2) )
      DIFDIS = DSQRT ( WPRED(2)**2+WPRED(3)**2 )
      ERR = IDINT ( TODIS / DIFDIS )
      IF ( ADJ1 .NE. 1 ) GO TO 430
            WRITE ( 6,1240 ) EAZ1D,EAZ1M,EAZ1S
            WRITE ( 6,1250 ) EAZ2D,EAZ2M,EAZ2S
            WRITE ( 6,1260 ) TODIS
            WRITE ( 6,1270 ) DIFDIS
            WRITE ( 6,1280 ) ERR
            DO 380 I = 1,COUNTA
               COBM1 ( I ) = 1.0D0
               COBM2 ( I ) = 0.0D0
               COBM3 ( I ) = 0.0D0
               IF ( I .GT. COUNTD ) GO TO 370
                  J = I
350               CONTINUE
                  IF ( J .EQ. COUNTA ) GO TO 360
                     COBM2 ( I ) = COBM2(I)+DELTAY(J)
                     COBM3 ( I ) = COBM3(I)+DELTAX(J)
                     J = J + 1
                  GO TO 350
360               CONTINUE
                  COBM2 ( I ) = COBM2 ( I )
                  COBM3 ( I ) = COBM3 ( I )*(-1.0D0)
370            CONTINUE
380         CONTINUE
            DO 390 I = 1,COUNTD
               J = COUNTA + I
               COBM1 ( J ) = 0.0D0
               COBM2 ( J ) = DSIN ( COAZR ( I ))
               COBM3 ( J ) = DCOS ( COAZR ( I ))
390         CONTINUE
            NT = COUNTA + COUNTD
            CALL CLSQ ( NT,COUNTA,COUNTD,STDA,STDD,WPRED,COBM1,
     *      COBM2,COBM3,BM,BMT,PMI,PIBT,VAR,CORAD,PW,PRINTD,DM,
     *      BPIM,QBPIM,STAND,VTM )
C
C
C     CORRECTED OBSERVED DATA
```

```
C
C
                  DO 400 I = 1,COUNTA
                     ANG ( I ) = ANG ( I ) + CORAD ( I )
                     CALL CRDMS ( ANG(I),DUM1D,DUM1M,DUM1S )
                     ANGD ( I ) = IDINT ( DUM1D )
                     ANGM ( I ) = IDINT ( DUM1M )
                     ANGS ( I ) = DUM1S
  400             CONTINUE
                  DO 410 I = 1,COUNTD
                     J = COUNTA + I
                     DIST ( I ) = DIST ( I ) + CORAD ( J )
  410             CONTINUE
C
C     PRINT CORRECTED OBSERVED DATA
C
                  WRITE ( 6,1660 )
                  WRITE ( 6,1670 )
                  WRITE ( 6,1680 )
                  WRITE ( 6,1690 )
                  WRITE ( 6,1700 )
                  WRITE ( 6,1710 ) (STNAME(1,K1),K1=1,5)
                  WRITE ( 6,1720 ) (STNAME(2,K1),K1=1,5),ANGD(1),ANGM(
     *            1),ANGS(1)
                  WRITE ( 6,1730 ) DIST ( 1 )
                  DO 420 I = 1,COUNTP
                     K2 = I + 1
                     WRITE ( 6,1720 ) (TRNAME(I,K1),K1=1,5),ANGD(K2),AN
     *               GM(K2),ANGS(K2)
                     WRITE ( 6,1730 ) DIST ( K2 )
  420             CONTINUE
                  WRITE ( 6,1720 ) (STNAME(3,K1),K1=1,5),ANGD(COUNTA),
     *            ANGM(COUNTA),ANGS(COUNTA)
                  WRITE ( 6,1710 ) (STNAME(4,K1),K1=1,5)
                  SUMDX = 0.0D0
                  SUMDY = 0.0D0
  430          CONTINUE
  440       CONTINUE
  450 CONTINUE
      IF ( OPT .EQ. 1 ) GO TO 480
         WRITE ( 6,1310 )
         WRITE ( 6,1320 ) EAZ1D,EAZ1M,EAZ1S
         SUMDX = EGRID ( 3 ) - CEGRID ( COUNTD )
         SUMDY = NGRID ( 3 ) - CNGRID ( COUNTD )
         DIFDIS = DSQRT ( SUMDX**2 + SUMDY**2 )
         WRITE ( 6,1330 ) DIFDIS
         WRITE ( 6,1000 )
         WRITE ( 6,1790 ) STAND
```

```
            WRITE ( 6,1800 )
            DO 460 I = 1,COUNTA
               CALL CRDMS ( VAR (I),DUM1D,DUM1M,DUM1S )
               DUMMY1 = DUM1D*3600.0D0 + DUM1M*60.0D0 + DUM1S
               WRITE ( 6,1810 ) I,DUMMY1
 460        CONTINUE
            WRITE ( 6,1820 )
            DO 470 I = 1,COUNTD
               K4 = COUNTA + I
               WRITE ( 6,1830 ) I,VAR(K4)
 470        CONTINUE
 480  CONTINUE
      WRITE ( 6,1000 )
C
C
C     END COMPUTE_ADJUST_BY_CONDITION_EQUATIONS
C
C
      STOP
C
 1000 FORMAT ( '1' )
 1010 FORMAT ( ///,23X,'KNOWN HORIZONTAL STATION POSITIONS' )
 1020 FORMAT (     23X,'**********************************', // )
 1030 FORMAT ( 31X,'UTM GRID COORDINATES',6X,'AZIMUTH   GRID AZIMUTH',/)
 1040 FORMAT ( 28X,'GRID NORTHING  GRID EASTING  FROM TO',4X,'CLOCKWISE'
     *,/ )
 1050 FORMAT ( 1X,'NO',3X,'NAME OF STATIONS',35X,'NO',3X,'NO',4X,'FROM N
     *ORTH', / )
 1060 FORMAT ( 32X,'METERS',8X,'METERS',14X,'D. M.  S.',/ )
 1070 FORMAT ( 1X,I2,1X,5A4,3X,F14.5,2X,F12.5,2X,I2,' ->',I2,1X,I3,1X,I2
     *,1X,F8.5,/ )
 1080 FORMAT ( ///,31X,'OBSERVED DATA' )
 1090 FORMAT (     31X,'*************',// )
 1100 FORMAT ( 29X,'OBSERVED',19X,'  GRID  ',/ )
 1110 FORMAT ( 6X,'NAME OF STATIONS',8X,'ANGLE',8X,'STD.',9X,'DISTANCE',
     *5X,'STD.',/ )
 1120 FORMAT ( 26X,'D. M.  S.', 9X,'S.',11X,'METERS',7X,'M.' )
 1130 FORMAT ( /,4X,5A4,/ )
 1140 FORMAT ( 4X,5A4,1X,I3,1X,I2,1X,F8.5,1X,F8.5 )
 1150 FORMAT ( 52X,F14.5,1X,F8.5 )
 1160 FORMAT ( 4X,5A4,/ )
 1170 FORMAT ( ///,17X,'CALCULATION OF OPEN TRAVERSE STATION POSITIONS')
 1180 FORMAT ( ///,15X,'PRECALCULATED OF CLOSED TRAVERSE STATION POSITIO
     *NS' )
 1190 FORMAT ( 15X,'**********************************************',
     */// )
 1200 FORMAT ( 17X,'******************************************',// )
 1210 FORMAT ( 2X,'1',1X,5A4,3X,F14.5,2X,F12.5,/ )
```

```
1220 FORMAT ( 57X,I2,' ->',I2,1X,I3,1X,I2,1X,F8.5,/ )
1230 FORMAT ( 1X,I2,1X,5A4,/ )
1240 FORMAT ( ///,2X,'TOTAL AZIMUTH ERROR',8X,'= ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1250 FOFMAT ( /,2X,'AZIMUTH ERROR PER STATIONS = ',I3,' D. ',I2,' M. ',
    *F8.5,' S. ' )
1260 FORMAT ( /,'  TOTAL GRID DISTANCE          = ',F16.5,' METERS.' )
1270 FORMAT ( /,2X,'DISTANCE ERROR',13X,'= ',F10.5,' METERS. ' )
1280 FORMAT ( /,2X,'ACCURACY',18X,' =   1:',I12 )
1290 FORMAT ( ///,2X,'CALCULATION OF CLOSED TRAVERSE STATION POSITIONS
    *BY LEAST-SQUARES ADJUSTMENT' )
1300 FORMAT (      2X,'**********************************************
    ****************************',/  )
1310 FORMAT ( ///,2X,'CHECK ROUND OFF ERROR ' )
1320 FORMAT ( /,2X,'AZIMUTH    = ',I3,' D. ',I2,' M. ',F8.5,' S. ' )
1330 FORMAT ( /,2X,'DISTANCE   = ',F10.5,' METERS. ' )
1340 FORMAT ( I1 )
1350 FORMAT ( I2 )
1360 FCRMAT ( F20.12 )
1370 FORMAT ( 5A4 )
1380 FORMAT ( 10X,'**************************************************
    ********' )
1390 FORMAT ( 10X,'*',57X,'*' )
1400 FORMAT ( 10X,'*',9X,'THIS IS A PROGRAM TO CALCULATION OF TRAVERSE'
    *,4X,'*'/,10X,'*',4X,'STATION POSITIONS. THERE ARE TWO OPTIONS IN T
    *HIS',5X,'*',/,10X,'*',4X,'PROGRAM.:-',43X,'*' )
1410 FORMAT ( 10X,'*',4X,'1 CALCULATION OF OPEN TRAVERSE STATION POSITI
    *ONS.',4X,'*' )
1420 FORMAT ( 10X,'*',4X,'2 CALCULATION OF CLOSED TRAVERSE STATION POSI
    *TIONS',3X,'*' )
1430 FORMAT (   /,28X,'WHAT IS YOUR OPTION ?' )
1440 FORMAT (   ,20X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/   )
1450 FORMAT ( //,16X,'THERE ARE TWO OPTIONS TO ENTER DATA.',/ )
1460 FORMAT (     14X,'1 ENTER DATA BY DATA FILE.',/ )
1470 FORMAT (14X,'2 ENTER DATA BY INTERACTIVE FROM TERMINAL KEYBOARD.')
1480 FORMAT ( /,21X,'PLEASE ENTER 1 OR 2 FOR YOUR OPTION',/ )
1490 FORMAT ( //,16X,'WHAT IS THE NAME OF KNOWN POSITION # ',I1,'.?')
1500 FORMAT ( /,12X,'THIS KNOWN POSITION WAS USED FOR THE FIRST BACKSIG
    *HT.')
1510 FORMAT ( /,12X,'THIS KNOWN POSITION WAS USED FOR THE LAST FORESIGH
    *T.')
1520 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE FIRST POSITION AT WHICH
    *',/,12X,'AN ANGLE WAS OBSERVED.' )
1530 FORMAT ( /,12X,'THIS KNOWN POSITION IS THE LAST POSITION AT WHICH'
    *,/,12X,'AN ANGLE WAS OBSERVED.' )
1540 FORMAT ( /,12X,'FOR EXAMPLE : GRID NORTHING 4,072,555.85206 METERS
    *.',/,26X,'ENTER 4072555.85206ODO' )
1550 FOFMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION OF THE OBSERVED AN
```

```
     *GLE',/,12X,'AT THIS STATION IN SECONDS ?',/,12X,'FOR EXAMPLE STD.
     *= 3.9 SECONDS. : ENTER 3.90D0 ' )
1560 FORMAT ( /,12X,'FOR EXAMPLE : GRID EASTING 608,279.04404 METERS.',
     */,26X,'ENTER 608279.04404D0' )
1570 FORMAT ( //,12X,'WHAT IS THE STANDARD DEVIATION IN METERS.?',/,12X
     *,'FOR EXAMPLE  STD. = .011 METERS : ENTER 0.011D0' )
1580 FORMAT ( /,12X,'HOW MANY POSITIONS DID YOU OBSERVE ANGLES ?',//,19
     *X,'PLEASE ENTER 01 TO 32' )
1590 FORMAT ( /,12X,'ENTER THE OBSERVED ANGLE AT THE FIRST KNOWN POSITI
     *ON.' )
1600 FORMAT ( /,10X,'ENTER THE OBSERVED ANGLE AT TRAVERSE STATION POSIT
     *ION # ',I2,'.' )
1610 FORMAT ( /,10X,'FOR EXAMPLE : OBSERVED ANGLE CLOCKWISE FROM BACKSI
     *GHT TO FORESIGHT',/,24X,'IS 169 D. 32 M. 11.11 S.',//,24X,'ENTER 1
     *69.321111D0' )
1620 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN THE FIRST KNOWN POSITI
     *ON AND',/,12X,'TRAVERSE STATION POSITION # 1.' )
1630 FORMAT ( /,12X,'ENTER GRID DISTANCE BETWEEN TRAVERSE STATION POSIT
     *ION # ',I2,'.',/,12X,'AND TRAVERSE STATION POSITION # ',I2,'.'
     * )
1640 FORMAT ( /,12X,'FOR EXAMPLE : THE GRID DISTANCE BETWEEN THE STATIO
     *NS IS',/,26X,'399.052 METERS.',//,26X,'ENTER 399.052D0' )
1650 FORMAT ( //,12X,'WHAT IS THE NAME OF TRAVERSE STATION POSITION # '
     *,I2,'.' )
1660 FORMAT ( ///,25X,'CORRECTED DATA' )
1670 FORMAT (       25X,'**************',// )
1680 FORMAT (  29X,'OBSERVED',9X,'  GRID  ',/ )
1690 FORMAT (   6X,'NAME OF STATIONS',8X,'ANGLE',11X,'DISTANCE',/ )
1700 FORMAT (  27X,'D. M.  S.',11X,'METERS',/ )
1710 FORMAT (   /,4X,5A4,/ )
1720 FORMAT (   4X,5A4,1X,I3,1X,I2,1X,F8.5 )
1730 FORMAT (  42X,F14.5 )
1740 FORMAT (  27X,'BY CONDITION EQUATION METHOD',// )
1750 FORMAT (  //,12X,'WHAT IS UTM GRID NORTHING AT THIS POSITION ? ' )
1760 FORMAT (  //,12X,'WHAT IS UTM GRID EASTING AT THIS POSITION ? ' )
1770 FORMAT (  //,16X,'WHAT IS UTM GRID AZIMUTH FROM',/,12X,5A4,' TO  ',
     *5A4 )
1780 FORMAT ( //,12X,'FOR EXAMPLE : UTM GRID AZIMUTH FROM  A.  TO  B.',
     */,26X,'IS 139 D. 19 M. 29.3 S. : ENTER 139.192930D0 ' )
1790 FORMAT (  //,1X,'THE STANDARD DEVIATION OF UNIT WEIGHT IS ',F15.10)
1800 FORMAT (  //,1X,'THE STANDARD DEVIATIONS OF ADJUSTED ANGLES ARE',/)
1810 FORMAT (      1X,'ANGLE # ',I2,' = ',F17.10,'  SECONDS',/ )
1820 FORMAT (  //,1X,'THE STANDARD DEVIATIONS OF ADJUSTED DISTANCES ARE'
     *,/ )
1830 FORMAT (     1X,'DISTANCE # ',I2,' = ',F16.10,'  METERS',/ )
C
      END
C
```

```
C
$EJECT
C                 CC WATFIV A1
C
C
      SUBROUTINE CLSQ ( NUMT,NA,ND,STDA,STDD,WM,BMR1,BMR2,BMR3,BM,BMT,PM
     *I,PIBT,VAR,VM,PW,PRINTD,DM,BPIM,QBPIM,STAND,VTM )
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C       AUTHOR : LT. SAMAN AUMCHANTR RTN.                             C
C         DATE :  JULY  4 , 1984                                      C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     P
C        THIS IS A SUBROUTINE FOR CALCULATION OF LEAST-SQUARES        P
C    ADJUSTMENT. ( CONDITION EQUATION METHOD )                        P
C                                                                     P
CPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
C                                                                     V
C                   VARIABLE DEFINITIONS                              V
C                                                                     V
C      WM     = PRECALCULATED DISCREPANCY ( W MATRIX )                V
C      BM     = B MATRIX ; CONSTANT COEFFICIENTS                      V
C      BMT    = THE TRANSPOSE OF B MATRIX                             V
C      PMI    = THE INVERSION OF P MATRIX                             V
C      KM     = K MATRIX ( LAGRANGE MULTIPLIER )                      V
C      VM     = V MATRIX ( THE CORRECTION VECTOR )                    V
C      USWFM  = SUBROUTINE FOR PRINT A MATRIX                         V
C      LINV2F = SUBROUTINE FOR INVERSION OF A MATRIX                  V
C      VMULFF = SUBROUTINE FOR MATRIX MULTIPLICATION                  V
C      NA     = THE TOTAL NUMBER OF OBSERVED ANGLE                    V
C      ND     = THE TOTAL NUMBER OF MEASURED DISTANCE                 V
C      NUMT   = THE SUMMATION OF THE TOTAL NUMBER OF OBSERVED ANGLES  V
C               AND MEASURED DISTANCE                                 V
C                                                                     V
CVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
C                                                                     C
C                    VARIABLE DECLARATIONS                            C
C                                                                     C
      DOUBLE PRECISION WM(3),BMR1(NUMT),BMR2(NUMT),BMR3(NUMT),BM(3,NUMT)
     *,BMT(NUMT,3),BBT(3,3),BBTI(3,3),WK3(18),KM(3),VM(NUMT),STDA(NA),ST
     *DD(ND),PMI(NUMT,NUMT),DUMMY1,DUMMY2,NUMO,VAR(NUMT),PIBT(NUMT,3),DM
     *(NUMT,NUMT),BPIM(3,NUMT),QBPIM(3,NUMT),STAND,VTM(1,NUMT)
C
      INTEGER I,J,PW,PRINTD,N1,IER,N3,IO5
C
      DATA NUMO/0.0D0/
```

```
C
C     COMPUTATION OF THE TRANSPOSE OF B MATRIX
C
      IO5 = 6
      N1 = 1
      N3 = 3
      DO  10 I = 1,NUMT
          BM ( 1,I ) = BMR1 ( I )
          BM ( 2,I ) = BMR2 ( I )
          BM ( 3,I ) = BMR3 ( I )
   10 CONTINUE
      DO  20 I = 1,3
          WM ( I ) = WM ( I ) * ( -1.0D0 )
   20 CONTINUE
      DO  40 I = 1,3
          DO  30 J = 1,NUMT
              BMT ( J,I ) = BM ( I,J )
   30     CONTINUE
   40 CONTINUE
C
C
C    INVERSION OF P MATRIX
C
C
      DO 70 I = 1,NUMT
         DO 60 J = 1,NUMT
            DM ( I,J ) = NUMO
            PMI ( I,J ) = NUMO
            IF ( I .NE. J ) GO TO 50
                 PMI ( I,J ) = 1.0D0
                 DM  ( I,J ) = 1.0D0
   50       CONTINUE
   60    CONTINUE
   70 CONTINUE
      IF ( PW .EQ. 1 ) GO TO 130
         DO 80 I = 1,NA
            DUMMY1 = STDA ( I )
            CALL CDMSR ( NUMO,NUMO,DUMMY1,DUMMY2 )
            VAP ( I ) = ( DSIN ( DUMMY2 ))**2
   80    CONTINUE
         DO 90 I = 1,ND
            J = NA + I
            VAR ( J ) = ( STDD ( I ))**2
   90    CONTINUE
         DO 120 I = 1,NUMT
            DO 110 J = 1,NUMT
               IF ( I .NE. J ) GO TO 100
                    PMI ( I,J ) = VAR ( I )
```

```
                DM  ( I,J ) = 1.0D0 / VAR ( I )
100          CONTINUE
110        CONTINUE
120      CONTINUE
130   CONTINUE
C
C
C     MULTIPLICATION OF INVERSION OF P MATRIX BY TRANSPOSE OF B MATRIX
C
C
      CALL VMULFF ( PMI,BMT,NUMT,NUMT,N3,NUMT,NUMT,PIBT,NUMT,IER )
C
C
C     MULTIPLICATION OF B MATRIX BY P.I*B.T MATRIX
C
      CALL VMULFF ( BM,PIBT,N3,NUMT,N3,N3,NUMT,BBT,N3,IER )
C
C     INVERSION OF B*B.T MATRIX
C
      CALL LINV2F ( BBT,N3,N3,BBTI,N1,WK3,IER )
C
C     COMPUTATION OF K MATRIX
C
      CALL VMULFF ( BBTI,WM,N3,N3,N1,N3,N3,KM,N3,IER )
C
C     COMPUTATION OF V MATRIX
C
      CALL VMULFF ( PIBT,KM,NUMT,N3,N1,NUMT,N3,VM,NUMT,IER )
C
C     THE CALCULATION OF THE STANDARD ERROR OF AN OBSERVATION WHICH
C     HAS UNIT WEIGHT
C
C
      DO 132 I = 1,NUMT
         VTM ( 1,I ) = VM ( I )
132   CONTINUE
      CALL VMULFF ( DM,VM,NUMT,NUMT,N1,NUMT,NUMT,VAR,NUMT,IER )
      CALL VMULFF ( VTM,VAR,N1,NUMT,N1,N1,NUMT,STAND,N1,IER )
C
C     THE CALCULATION OF B*P.I MATRIX
C
      CALL VMULFF ( BM,PMI,N3,NUMT,NUMT,N3,NUMT,BPIM,N3,IER )
C
C     THE CALCULATION OF (INVERSE B*P.I*B.T)*B*P.I MATRIX
C
      CALL VMULFF ( BBTI,BPIM,N3,N3,NUMT,N3,N3,QBPIM,N3,IER )
C
C     THE CALCULATION OF P.I*B.T*(INVERSE B*P.I*B.T)*B*P.I MATRIX
```

```
C
      CALL VMULFF ( PIBT,QBPIM,NUMT,N3,NUMT,NUMT,N3,DM,NUMT,IER )
C
C     PRINT DETAIL OF THE COMPUTATION
C
      IF ( PRINTD .EQ. 0 ) GO TO 150
         WRITE ( 6,1000 )
         WRITE ( 6,1010 )
         DO 140 I = 1,3
            WRITE ( 6,1020 ) WM ( I )
 140     CONTINUE
         WRITE ( 6,1030 )
         CALL USWFM ( 'R-C',N3,PMI,NUMT,NUMT,NUMT,IO5 )
         WRITE ( 6,1040 )
         CALL USWFM ( 'R-C',N3,BM,N3,N3,NUMT,IO5 )
         WRITE ( 6,1000 )
         WRITE ( 6,1050 )
         CALL USWFM ( 'R-C',N3,BMT,NUMT,NUMT,N3,IO5 )
         WRITE ( 6,1060 )
         CALL USWFM ( 'R-C',N3,PIBT,NUMT,NUMT,N3,IO5 )
         WRITE ( 6,1070 )
         CALL USWFM ( 'R-C',N3,BBT,N3,N3,N3,IO5 )
         WRITE ( 6,1000 )
         WRITE ( 6,1080 )
         CALL USWFM ( 'R-C',N3,BBTI,N3,N3,N3,IO5 )
         WRITE ( 6,1090 )
         CALL USWFM ( 'P-C',N3,KM,N3,N3,N1,IO5 )
         WRITE ( 6,1100 )
         CALL USWFM ( 'R-C',N3,VM,NUMT,NUMT,N1,IO5 )
         WRITE ( 6,1000 )
         WRITE ( 6,1110 )
         CALL USWFM ( 'R-C',N3,BPIM,N3,N3,NUMT,IO5 )
         WRITE ( 6,1000 )
         WRITE ( 6,1120 )
         CALL USWFM ( 'R-C',N3,QBPIM,N3,N3,NUMT,IO5 )
         WRITE ( 6,1000 )
         WRITE ( 6,1130 )
         CALL USWFM ( 'R-C',N3,DM,NUMT,NUMT,NUMT,IO5 )
         WRITE ( 6,1140 )
         WRITE ( 6,1020 ) STAND
C
 150  CONTINUE
C
C     THE CALCULATION OF P.I - P.I*B.T*(INVERSE B*P.I*B.T)*B*P.I MATRIX
C
      STAND = DSQRT ( DABS ( STAND / 3.0D0 ))
      DO 190 I = 1,NUMT
         DO 180 J = 1,NUMT
```

```
            DM ( I,J ) = PMI ( I,J ) - DM ( I,J )
            IF ( I .NE. J ) GO TO 170
                VAR ( I ) = STAND*(DSQRT(DABS(DM(I,J))))
  170       CONTINUE
  180     CONTINUE
  190   CONTINUE
        IF ( PRINTD .EQ. 0 ) GO TO 200
            WRITE ( 6,1000 )
            WRITE ( 6,1150 )
            CALL USWFM ( 'R-C',N3,DM,NUMT,NUMT,NUMT,IO5 )
  200   CONTINUE
        WRITE ( 6,1000 )
        RETURN
C
C
C
 1000 FORMAT ( '1' )
 1010 FORMAT ( ///,13X,'-W MATRIX' ,/ )
 1020 FORMAT ( /,12X,F15.10 )
 1030 FORMAT ( ///,35X,'INVERSION OF P MATRIX ',// )
 1040 FORMAT ( ///,40X,'B MATRIX',// )
 1050 FORMAT ( ///,17X,'TRANSPOSE OF B MATRIX',// )
 1060 FORMAT ( ///,22X,'P.I*B.T MATRIX',// )
 1070 FORMAT ( ///,21X,'B*P.I*B.T MATRIX',// )
 1080 FORMAT ( ///,15X,'INVERSION OF B*P.I*B.T MATRIX',// )
 1090 FORMAT ( ///, 7X,'K MATRIX',// )
 1100 FORMAT ( ///, 7X,'V MATRIX',// )
 1110 FORMAT ( ///,5X,'B*P.I MATRIX',/ )
 1120 FORMAT ( ///,5X,'(INVERSE B*P.I*B.T)*B*P.I MATRIX',/ )
 1130 FORMAT ( ///,5X,'P.I*B.T*(INVERSE B*P.I*B.T)*B*P.I MATRIX',/ )
 1140 FORMAT ( ///,5X,'V.T*P*V MATRIX',/ )
 1150 FORMAT ( ///,5X,'P.I - P.I*B.T*(INVERSE B*P.I*B.T)*B*P.I MATRIX',
     */ )
C
C
        END
C
C
```

LIST OF REFERENCES

1. Bomford, Brigadier G., *Geodesy*, Second Edition, p. 1, Oxford University Press, 1962.

2. Palikaris, Athanasios E., *Methods of Hydrographic Surveying used by Different Countries*, M.S. Thesis, Naval Postgraduate School, Monterey, California, p. 139, 1983.

3. Davis, Raymond E., Foote, Francis S., Anderson, James M., and Mikhail, Edward M., *Surveying Theory and Practice*, Sixth Edition, McGraw-Hill, 1981.

4. Department of the Army Technical Manual (TM 5-237), *Surveying Computer's Manual*, 1964.

5. U.S. Department of Commerce, Federal Geodetic Control Committee, *Classification, Standards of Accuracy, and General Specifications of Geodetic Control Surveys*, pp. 6-7, 1974.

6. U.S. Department of Commerce, Environmental Science Services Administration, Coast and Geodetic Survey, *Horizontal Control Data*, Quad 361214, Stations 1037, 1058, 1059A, and 1068, 1963.

7. Mikhail, Edward M., and Gracie, Gordon, *Analysis and Adjustment of Survey Measurements*, Van Nostrand Reinhold Company, 1981.

8. Hardy, Rolland L., *A Brief Outline and Demonstration of the Theory and Practice of Least Squares*, p. 1, Unpublished Note.

9. Hardy, Rolland L., *Least Squares Adjustment of Traverse*, p. 4, Unpublished Note.

10. Pacific Marine Center, NOS, *Horizontal Control Report for Moss Landing - Naval Postgraduate School Survey*, 1983.

BIBLIOGRAPHY

Durgin, Casper M. and Sutcliffe, Walter D., _Manual of First-order Traverse_, Coast and Geodetic Survey Special Publication No. 137, U.S. Department of Commerce, 1927.

Ewing, Clair E. and Mitchell, Michael M., _Introduction to Geodesy_, Elsevier North Holland Publishing Company, 1969.

Graham and Neil, _Introduction to Computer Science A Structure Approach_, West Publishing Company, 1982.

Hodgson, C.V., _Manual of Second and Third Order Triangulation and Traverse_, Coast and Geodetic Survey Special Publication No. 145, U.S. Department of Commerce, 1935.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library Code 0142 Naval Postgraduate School Monterey, CA 93943	2
3.	Chairman, Department of Oceanography Code 68Mr Naval Postgraduate School Monterey, CA 93943	1
4.	NOAA Liaison Officer Post Office Box 8688 Monterey, CA 93943	1
5.	Office of the Director Naval Oceanography Division (OP-952) Department of the Navy Washington, D.C. 20350	1
6.	Commander Naval Oceanography Command NSTL, MS 39529	1
7.	Commanding Officer Naval Oceanographic Office Bay St. Louis NSTL, MS 39522	1
8.	Commanding Officer Naval Ocean Research and Development Activity Bay St. Louis NSTL, MS 39522	1
9.	Chief of Naval Research 800 North Quincy Street Arlington, VA 22217	1
10.	Chairman, Oceanography Department U. S. Naval Academy Annapolis, MD 21402	1
11.	Chief, Hydrographic Programs Division Defense Mapping Agency (Code PPH) Building 56 U.S. Naval Observatory Washington, D. C. 20305-3000	1
12.	Lt. Saman Aumchantr Hydrographic Department Royal Thai Navy Bangkok 10600 THAILAND	2

13.	Personnel Department Royal Thai Navy Bangkok 10600 THAILAND	1
14.	Library Hydrographic Department Royal Thai Navy Bangkok 10600 THAILAND	2
15.	Chief, Hydrographic Surveys Branch N/CG24, Room 404, WSC-1 National Oceanic and Atmospheric Administration Rockville, MD 20852	1
16.	Program Planning, Liaison, and Training Division NC2, Room 105, Rockwall Building National Oceanic and Atmospheric Administration Rockville, MD 20852	1
17.	Director, Pacific Marine Center N/MOP National Ocean Service, NOAA 1801 Fairview Avenue, East Seattle, WA 98102	1
18.	Director, Atlantic Marine Center N/MOA National Ocean Service, NOAA 439 West York Street Norfolk, VA 23510	1
19.	Dr. Rolland L. Hardy Department of Oceanography, Code 68 Xz Naval Postgraduate School Monterey, CA 93943	2
20.	Associate Deputy Director for Hydrography Defense Mapping Agency (Code DH) Building 56 U.S. Naval Observatory Washington, D.C. 20305-3000	1

END

FILMED

6-85

DTIC